A gentle introduction to the PCP theorem, part 2

Olivier Bégassat

Contents

Recap

In the first post in this series we gave basic definitions : the complexity classes **NP** and **PCP** and what it takes for a decision problem to have probabilistically checkable proofs (PCPs), and thus lie in a PCP complexity class. We also teased the statement of the weak PCP theorem:

$$\mathbf{NP} \subset \mathbf{PCP}_{1,\frac{1}{2}}\Big[O(\text{poly}(n)), O(1)\Big].$$

where

$$\mathbf{PCP}_{1,\frac{1}{2}}\Big(\text{poly}(n), O(1)\Big) = \bigcup_{c,C,Q\geq 1} \mathbf{PCP}_{1,\frac{1}{2}}\Big(n^c + C, Q\Big)$$

In this second post we sketch a proof of this result. We warn the reader that proving it requires a bit of linear algebra and (unsurprisingly) some basic probability theory. But the general gist of the proof isn't all that complicated (although it *is* ingenious). As is usual in these kinds of proofs, the hard part is establishing *soundness*. We will provide most of the proof in what follows, everything except for a result from discrete Fourier theory. The proof is taken from this excellent Introduction to the PCP theorem by Min Jae Song, where the reader will also find the proof of the aforementioned result.

To simplify notations, write **PCP** for the complexity class $\mathbf{PCP}_{1,\frac{1}{2}}(O(\ln(n)), 0(1))$ that appears in the statement of the strong PCP theorem and **wPCP** for the complexity class $\mathbf{PCP}_{1,\frac{1}{2}}(\text{poly}(n), 0(1))$ that appears in the statement of the weak PCP theorem.

Where we're headed

There will be a lot of ground to cover, so we shall divide the work up into smaller chunks and provide the reader with a bird eye's view of what's to come.

Pre-processing

The bulk of this post will be about constructing PCPs for **3-SAT** using the decision problem known as **QUADEQ** as an intermediary. We will thus see how an to reduce **3-SAT** to **QUADEQ** (along the way we shall of course describe **QUADEQ**, i.e. its instances and witnesses). This first transformation from **3-SAT** to **QUADEQ** is very lean: hardly any complexity gets added. In doing so, we move from a typical computer science problem to one that has more of a *linear algebra* flavour to it.

The next step is again to change the problem to an equivalent one, however, this transformation comes at a **tremendous cost**: the solution/witness space grows exponentially larger in size. In other words, this modification brutally impacts the witness / PCP proof-format size for **QUADEQ**.

The Prover's job and the PCP proof format

Thankfully, this proof-format is quite concrete and explicit: rather than a 0/1-assignment as in **3-SAT** or a $\{0,1\}$-vector for **QUADEQ**, proofs in the PCP format are pairs of $\{0,1\}$-valued functions f and g given by their values / graphs Γ_f and Γ_g (very large arrays of bits). Correct proofs are special functions (*linear* functions) (Walsh-Hadamard codes) that are *derived from a witness*. The series of conversions

$$(\mathbf{3-SAT}\text{-witness}) \Rightarrow (\mathbf{QUADEQ}\text{-witness}) \Rightarrow (\mathbf{QUADEQ}\text{ PCP proof})$$

is handled by the prover and involves only trivial computations (dot products), albeit a great many of them. It requires as only input a witness u of the specific **3-SAT** instance, and cannot be precomputed. In fact, it involves so many computations that producing such PCP proofs for instances of **3-SAT** with 10 boolean variables already is totally out of the question. We stress that this is not meant to be a practical PCP scheme. At this point we will have described the proof format, and how a prover contructs a proof (Γ_f, Γ_g) from witness data u.

The Verifier's job

The next step is describing the job of the verifier. As promised, its job will be

1. to toss a polynomial amount of random coins,
2. to perform simple polynomial time computations using on bits, and outputting either 1 or 0.

The purpose of the random coin tosses is to come up with random locations within Γ_f and Γ_g to look up. Thus the verifier reads a constant number of randomly (and independently) sampled bits $(a, b, c, d, e, A, B, C, D, E, F, G$ below) from the proof (Γ_f, Γ_g) and uses them as inputs for computations. To give the reader an idea of just how simple the actual computations are, we list them here:

1. fetch $a, b, c \longleftarrow \Gamma_f$, check if $a + b \stackrel{?}{=} c$;
2. fetch $A, B, C \longleftarrow \Gamma_g$, check if $A + B \stackrel{?}{=} C$;
3. fetch $d, e \longleftarrow \Gamma_f$ and $D, E \longleftarrow \Gamma_g$ and check $d \cdot e \stackrel{?}{=} D + E$;
4. fetch $F, G \longleftarrow \Gamma_g$ and $\vec{x} \longleftarrow \mathbb{F}_{\not\varepsilon}{}^m$, compute a dot product $p = \langle \vec{b} \mid \vec{x} \rangle$ check and if $F + G \stackrel{?}{=} p$.

Of course there is some structure to these queries which we gloss over for now, but we insist on the fact that the actual computations are incredibly tame. To be precise, operations of type 1. , 2., 3. and the check of 4. are constant time operations on "bits" $a, b, c, d, e, A, B, C, D, E, F, G \in \{0,1\}$ to be repeated Q times (so constant cost), and computing the dot product $\langle \vec{b} \mid \vec{x} \rangle$ has linear complexity in the vector size.

Testing and Soundness

Good "tests" are the central pillar of the **PCP** verifier, and so we expound a simple view on "testing". Properties of linear and bilinear maps are front and center in the proof of the weak PCP theorem. We'll recall a couple of standard facts from linear algebra which explain the benefits of the previous modifications. The core benefit is that this new proof format brings **uniform separation** to the space of solutions where previously potential solutions could be arbitrarily close.

It will be quite clear from that construction that proofs which were correctly constructed by an honest verifier in possession of corredt witness data have an acceptance rate of 100%, so **Completeness** is a given. Where things really get interesting is with **Soundness**, i.e. in showing that a dishonest prover can't come up with fake proofs for **NO** instances of **QUADEQ** that would get accepted $\geq 50\%$ of the time. This requires some real ingenuity and criteria for (probabilistically) recognizing the graphs of linear functions, see the section on Linearity testing, and this is when the testing we talked about earlier becomes important. Careful analysis of this protocol shows that this verifier has the soundness required of a weak PCP verifier, and so **3-SAT** has PCPs. But **3-SAT** is but one decision problem in the infinite collection of decision problems that is **NP** : do we have to start all over for other decision problems in **NP**? No if **wPCP** contains one NP complete problem, then it contains all of **NP**. Since **3-SAT** (and **QUADEQ** for that matter) is NP-complete, the weak PCP theorem is proven :)

Algebraisation of 3-SAT: from 3-SAT to QUADEQ

> We describe how to convert an instance of **3-SAT**, a decision problem from the realm of boolean algebra, into an equivalent problem formulated in terms of linear algebra over the field with two elements $\mathbb{F}_2$.

Algebra over the field $\mathbb{F}_2$.

All the algebra will happen over the field with two elements $\mathbb{F}_2$. We quickly recall its arithmetic. As a set $\mathbb{F}_2 = \{0, 1\}$. Addition and a multiplication are defined by

$+$	0	1
0	0	1
1	1	0

$\times$	0	1
0	0	0
1	0	1

Note that for all $x \in \mathbb{F}_2$, $x^2 = x$. The arithmetic of $\mathbb{F}_2$ is that of the usual integers if all we care about is parity, i.e. even vs. odd numbers. Indeed, we can pass from the integers $\mathbb{Z}$ to $\mathbb{F}_2$ by sending an arbitrary integer $n \in \mathbb{Z}$ to $p(n) = 0$ if n is even, and to $p(n) = 1$ if n is odd. Then, since the sum of two integers is even *iff* the two integers have the same parity and the product of two integers is odd *iff* both are odd, we see that for arbitrary integers n, m

$$p(n + m) = p(n) + p(m) \quad \text{and} \quad p(n \times m) = p(n) \times p(m)$$

where the right-hand side + and × are those of $\mathbb{F}_2$.

Boolean algebra from the $\mathbb{F}_2$-point of view

We can express all boolean operations by using the field operations + and × of $\mathbb{F}_2$.

The correspondence between basic boolean operations and algebraic operations is contained in the following table:

x	y	$1+x$	NOT x	$x+y$	x XOR y	xy	x AND y	$x+y+xy$	x OR y
0	0	1	1	0	0	0	0	0	0
0	1	1	1	1	1	0	0	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	0	0	0	1	1	1	1

3-SAT from the $\mathbb{F}_2$-point of view

As a consequence of the previous paragraph, satisfiability of a boolean formula φ is equivalent to the satisfiability of a system of polynomial equations in $\mathbb{F}_2$.

We use standard notation: $\neg\ x$ stands for NOT x, $x\ \vee\ y$ stands for x OR y and $x\ \wedge\ y$ stands for x AND y. Recall that an instance of **3-SAT** is a boolean formula φ in 3-conjunctive normal form, that is, φ comes with a special representation as a conjunction of clauses of the form $x\ \vee\ y\ \vee\ z$, $x\ \vee\ y\ \vee\ \neg z$, $x\ \vee\ \neg y\ \vee\ \neg z$ or $\neg x\ \vee\ \neg y\ \vee\ \neg z$, where the place holder-variables x, y, z are replaced with 3 variables chosen from $x_1, \ldots, x_n$:

$$\varphi(x_1, \ldots, x_n) = \bigwedge_{i=1}^{c} (y_1^i \vee y_2^i \vee y_3^i)$$

Here c is the number of clauses involved and the boolean variables $y_1^1, y_2^1, y_3^1, \ldots, y_1^c, y_2^c, y_3^c$ are taken from $\{x_1, x_2, \ldots, x_n, \neg x_1, \neg x_2, \ldots, \neg x_n\}$. Recall furthermore that φ is said to be *satisfiable* if there exists an assignment $u_1, \ldots, u_n \in \{0, 1\}$ such that $\varphi(u_1, \ldots, u_n) = 1$.

$$\varphi(u_1, \ldots, u_n) = 1 \quad \Longleftrightarrow \quad \begin{cases} y_1^1 \vee y_2^1 \vee y_3^1 = 1 \\ y_1^2 \vee y_2^2 \vee y_3^2 = 1 \\ \qquad\qquad\qquad\ \vdots \\ y_1^n \vee y_2^n \vee y_3^n = 1 \end{cases}$$

Let us define the complexity of φ to be $c + n$.

We can replace the condition $x\ \vee\ y\ \vee\ z = 1$ for three boolean unknowns with a system of two quadratic equations in $\mathbb{F}_2$. The first line introduces an auxiliary boolean variable t defined to be logically equivalent to $x \vee y$.

$$\begin{cases} x + y + xy + t = 0 & : \quad t \iff x \vee y \\ t + z + tz = 1 & : \quad x \vee y \vee z = 1 \end{cases}$$

the first quadratic equation introduces an auxiliary boolean variable t defined to be logically equivalent to $x \vee y$. The second one is logically equivalent to t OR $z = \mathsf{TRUE}$, i.e. $x \vee y \vee z = \mathsf{TRUE}$. (Note that if we didn't want to introduce the extra variable t we could have expressed everything in terms of the *cubic equation* $x \vee y \vee z = 1$ as $x + y + z + yz + zx + xy + xyz = 1$.)

$$\varphi(u_1, \ldots, u_n) = \bigwedge_i^c (y_1^i \vee y_2^i \vee y_3^i) = 1 \iff \begin{cases} y_1^1 \vee y_2^1 \vee y_3^1 = 1 \\ y_1^2 \vee y_2^2 \vee y_3^2 = 1 \\ \vdots \\ y_1^c \vee y_2^c \vee y_3^c = 1 \end{cases}$$

(All y_j^i are either u_k or $\neg\, u_k$)

$$\begin{cases} y_1^1 + y_2^1 + y_1^1 y_2^1 + t^1 = 0 \\ y_3^1 + t^1 + y_3^1 t^1 = 1 \\ y_1^2 + y_2^2 + y_1^2 y_2^2 + t^2 = 0 \\ y_3^2 + t^2 + y_3^2 t^2 = 1 \\ \vdots \\ y_1^c + y_2^c + y_1^c y_2^c + t^c = 0 \\ y_3^c + t^c + y_3^c t^c = 1 \end{cases}$$

In other words, the satisfiability of the boolean formula φ is equivalent to the satisfiability in $\mathbb{F}_2$ of a certain system of $2c$ quadratic equations in $n + c$ variables.

QUADEQ

> **QUADEQ** is a decision problem generalizing the sort of system of quadratic equations in $\mathbb{F}_2$ defined in the previous section.

Having observed that the satisfiability of a boolean formula φ in 3-CNF is equivalent to the simultaneous satisfiability of a system of quadratic equations in $\mathbb{F}_2$, we are ready to give the definition of the decision

problem **QUADEQ**.

- An instance of **QUADEQ** consists of m quadratic equations with coefficients in $\mathbb{F}_2$ and with n unknowns $u_1, \dots, u_n$ in $\mathbb{F}_2$
- YES instances of **QUADEQ** are those systems of quadratic equations for which there exists an assignment of the varibles $u_1, \dots, u_n \in \mathbb{F}_2$ that satisfies all m quadratic equations simultaneously.

We think of the assignment $u_1, \dots, u_n$ as a vector $u = {}^t(u_1, \dots, u_n) \in \mathbb{F}_2^n$. Here's an instance of **QUADEQ** with $m = 5$ quadratic equations and $n = 4$ boolean variables :

$$\begin{cases} u_1 + u_1u_4 + u_2u_3 + u_3u_4 + u_4 = 1 \\ u_1u_2 + u_2 + u_2u_4 + u_3 + u_4 = 0 \\ u_1u_3 + u_2 + u_2u_3 + u_3 = 1 \\ u_1 + u_1u_3 + u_1u_4 + u_2u_3 + u_2u_4 + u_3 + u_3u_4 + u_4 = 1 \\ u_1u_4 + u_2u_4 + u_3 + u_3u_4 = 1 \end{cases}$$

This instance of **QUADEQ** has a solution given by $(u_1, u_2, u_3, u_4) = (0, 1, 1, 1)$. Notice that in the previous case, every quadratic equation only mixed 4 variables, this restriction is lifted in **QUADEQ**.

It is clear that the decision problem **QUADEQ** lies in **NP** : to convince a verifier that a system of m boolean quadratic equations such as above has a solution (i.e. is a "YES" instance of **QUADEQ**) it is enough to provide the n bits of a satisfying assignement. Verification time is polynomial time in m and n. Furthermore, in the previous section we described a polynomial time reduction from **3-SAT** to **QUADEQ**. It follows that **QUADEQ** is NP-complete.

QUADEQ with matrices

Going back to the example instance of **QUADEQ**, we can express it more systematically as

$$\begin{cases} 1 \cdot u_1 + 0 \cdot u_1u_2 + 0 \cdot u_1u_3 + 1 \cdot u_1u_4 + 0 \cdot u_2 + 1 \cdot u_2u_3 + 0 \cdot u_2u_4 + 0 \cdot u_3 + 1 \cdot u_3u_4 + 1 \cdot u_4 = 1 \\ 0 \cdot u_1 + 1 \cdot u_1u_2 + 0 \cdot u_1u_3 + 0 \cdot u_1u_4 + 1 \cdot u_2 + 0 \cdot u_2u_3 + 1 \cdot u_2u_4 + 1 \cdot u_3 + 0 \cdot u_3u_4 + 1 \cdot u_4 = 0 \\ 0 \cdot u_1 + 0 \cdot u_1u_2 + 1 \cdot u_1u_3 + 0 \cdot u_1u_4 + 1 \cdot u_2 + 1 \cdot u_2u_3 + 0 \cdot u_2u_4 + 1 \cdot u_3 + 0 \cdot u_3u_4 + 0 \cdot u_4 = 1 \\ 1 \cdot u_1 + 0 \cdot u_1u_2 + 1 \cdot u_1u_3 + 1 \cdot u_1u_4 + 0 \cdot u_2 + 1 \cdot u_2u_3 + 1 \cdot u_2u_4 + 1 \cdot u_3 + 1 \cdot u_3u_4 + 1 \cdot u_4 = 1 \\ 0 \cdot u_1 + 0 \cdot u_1u_2 + 0 \cdot u_1u_3 + 0 \cdot u_1u_4 + 0 \cdot u_2 + 0 \cdot u_2u_3 + 1 \cdot u_2u_4 + 1 \cdot u_3 + 1 \cdot u_3u_4 + 0 \cdot u_4 = 1 \end{cases}$$

In general, to specify an instance of **QUADEQ** with m equations and n unknowns $u_1, \dots, u_n$ we may thus specify, for each of the m equations, coefficients $a_{i,j}^k \in \mathbb{F}_2 = \{0, 1\}$ which record which of the n^2 products u_iu_j with $1 \leq i, j \leq n$ will make an appearance in a given line k, aswell as target values :

$$\begin{cases} a_{1,1}^1 \cdot u_1u_1 + a_{1,2}^1 \cdot u_1u_2 + \cdots + a_{1,n}^1 \cdot u_1u_n + \bullet\bullet\bullet + a_{n,1}^1 \cdot u_nu_1 + a_{n,2}^1 \cdot u_nu_2 + \cdots + a_{n,n}^1 \cdot u_nu_n = b_1 \\ a_{1,1}^2 \cdot u_1u_1 + a_{1,2}^2 \cdot u_1u_2 + \cdots + a_{1,n}^2 \cdot u_1u_n + \bullet\bullet\bullet + a_{n,1}^2 \cdot u_nu_1 + a_{n,2}^2 \cdot u_nu_2 + \cdots + a_{n,n}^2 \cdot u_nu_n = b_2 \\ \vdots \qquad\qquad \vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots \qquad\qquad\qquad\qquad \vdots \\ a_{1,1}^m \cdot u_1u_1 + a_{1,2}^m \cdot u_1u_2 + \cdots + a_{1,n}^m \cdot u_1u_n + \bullet\bullet\bullet + a_{n,1}^m \cdot u_nu_1 + a_{n,2}^m \cdot u_nu_2 + \cdots + a_{n,n}^m \cdot u_nu_n = b_m \end{cases}$$

There is a lot of redundancy : $u_i^2 = u_i$ and $u_iu_j = u_ju_i$, but we keep all these products and store them in the vector $\mathbf{U} = u \otimes u \in \mathbb{F}_2^n \otimes \mathbb{F}_2^n \simeq \mathbb{F}_2^{n^2}$. Only when doing tests will we ever need the fact $\mathbf{U} = u \otimes u \in \mathbb{F}_2^{n^2}$ lives most naturally in the tensor product $\mathbb{F}_2^n \otimes \mathbb{F}_2^n$ (but for now this is irrelevant). If we collect all the coefficients $a_{i,j}^k$ in a matrix, and all the target values b_i into a vector, this system can be expressed matricially as

Equivalent problem statement :

$$\text{Find } u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \text{ s.t. } A\mathbf{U} = B \text{ where } A \text{ is a matrix, } \mathbf{U} = u \otimes u = \begin{pmatrix} u_1u_1 \\ u_1u_2 \\ \vdots \\ u_1u_n \\ u_2u_1 \\ u_2u_2 \\ \vdots \\ u_2u_n \\ \vdots \\ u_nu_1 \\ u_nu_2 \\ \vdots \\ u_nu_n \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}.$$

Where A

$$A = (a^k_{i,j})_{\substack{1 \le k \le m \\ 1 \le i,j \le n}} \in \mathcal{M}_{m,n^2}(\mathbb{F}_2)$$

is the $m \times n^2$ dimensional matrix $(a^k_{i,j})_{\substack{1 \le k \le m \\ 1 \le i,j \le n}}$, $X = (x_i x_j)_{1 \le i \le j \le n}$ and $B = {}^t(b_1, \ldots, b_m)$ is the vector of target values.

Modifying the witness / proof format of QUADEQ

We described the first part of the transformation, the lean part. Witness-size hardly changed when moving from $\{0,1\}$-assignments witnessing for **3-SAT** to vectors ${}^t(u_1, \ldots, u_p) \in \mathbb{F}_2^p$ witnessing for **QUADEQ**. The next part of the transformation is the costly part. The central idea is that we may replace vectors $w \in \mathbb{F}_2^p$ (say **QUADEQ** witnesses) with (linear) maps $\langle w \mid - \rangle : \mathbb{F}_2^p \to \mathbb{F}_2$. We will further treat these (linear) maps as if they were ordinary maps $\mathbb{F}_2^p \to \mathbb{F}_2$.

The costlyness of this alternate (but equivalent) description of vectors is apparent: a vector in $\mathbb{F}_2^p$ is p bits, a function $\mathbb{F}_2^p \to \mathbb{F}_2$ is 2^p bits.

Dot products and replacing vectors by maps

We can safely replace vectors $w \in \mathbb{F}_2^p$ with linear maps $\langle w \mid - \rangle : \mathbb{F}_2^p \to \mathbb{F}_2$. In this paragraph we explain how to replace a vector $w \in \mathbb{F}_2^p$ with an ordered list of bits $\Gamma_{\langle w|-\rangle}$ of length 2^p.

We use the standard notation for dot products

$$\langle x \mid y \rangle = \sum_{i=1}^{p} x_i y_i$$

for vectors $x = {}^t(x_1, \ldots, x_p), y = {}^t(y_1, \ldots, y_p) \in \mathbb{F}_2^p$.

We need an elementary lemma from linear algebra. Let $p \geq 1$ be a positive integer. Recall that a *linear map* $\varphi : \mathbb{F}_2^p \longrightarrow \mathbb{F}_2$ is a map such that for all vectors $x, y \in \mathbb{F}_2^p$, $\varphi(x+y) = \varphi(x) + \varphi(y)$.

- Two vectors $u, v \in \mathbb{F}_2^p$ are the same if and only if for every vector $x \in \mathbb{F}_2^p$, $\langle u \mid x \rangle = \langle v \mid x \rangle$.
- Let $u \in \mathbb{F}_2^p$ be a vector, then the map $\langle u \mid - \rangle : (\mathbb{F}_2)^p \to \mathbb{F}_2,\ x \mapsto \langle u \mid x \rangle$ is linear.
- Conversely, for every linear map $\varphi : (\mathbb{F}_2)^p \to \mathbb{F}_2$ there exists a unique vector $u_\varphi \in (\mathbb{F}_2)^p$ such that $\varphi = \langle u_\varphi \mid - \rangle$.

In other words, no information is lost when moving from a vector u to the associated linear map $\langle u \mid - \rangle$, and every linear map $\mathbb{F}_2)^p \to \mathbb{F}_2$ arises uniquely in just that way.

Let us do a concrete example of this. Let us take $p = 3$, and $u = (0,1,1) \in \mathbb{F}_2^3$. We can write down the set of tuples $(x; \langle u \mid x \rangle)$ where x runs through all the points of $\mathbb{F}_2^3$:

$$\Gamma_{\langle u|-\rangle} \equiv \left\{ \begin{array}{l} (0,0,0;0), (0,0,1;1), (0,1,0;1), (0,1,1;0), \\ (1,0,0;0), (1,0,1;1), (1,1,0;1), (1,1,1;0) \end{array} \right\}$$

If we agree to order the vectors of $\mathbb{F}_2^3$ lexicographically, i.e. in the order $\mathbb{F}_2^3 = \{(0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1$ we can dispense with the coordinates of x and simply write down the ordered list of the $\langle u \mid x \rangle$:

$$\Gamma_{\langle u|-\rangle} \equiv [0,1,1,0,0,1,1,0]$$

This graph takes up 2^3 bits of information.

More generally we can encode any vector $w \in \mathbb{F}_2^p$ as a map $\langle w \mid - \rangle : \mathbb{F}_2^p \to \mathbb{F}_2$, and if we order the vectors of $\mathbb{F}_2^p$ lexicographically, we can encode the graph $\Gamma_{\langle w|-\rangle}$ as an ordered list of bits of length 2^p. Note that there are 2^{2^p} maps $\mathbb{F}_2)^p \to \mathbb{F}_2$, only 2^p of which are linear. In other words, amongst all maps, *linear* maps represent a tiny minority, they are needles in an exponential haystack. Also, this encoding is *incredibly wasteful*: to specify a vector in $\mathbb{F}_2^p$ we only need p bits (its p coordinates), specifying its encoding uses 2^p bits.

(Note: the term `graph` from now on refers to graphs of functions.)

Adapting QUADEQ to accept maps as witnesses

> **QUADEQ**, as we defined it, accepts *vectors* as witnesses. Since vectors can be encoded as maps, we consider a variant **QUADEQ** of **QUADEQ** whose instances are the same as those of **QUADEQ** (i.e. pairs (A, b) where A is a matrix and b is vector) but whose witnesses are *maps* given by their values Γ_f. We can upgrade this nonsense to the decision problem **QUADEQ** with the same instances as **QUADEQ**, but whose witnesses are *pairs of maps* given by their values Γ_f and Γ_g. Both variants of **QUADEQ** have precisely the same YES instances as **QUADEQ**. From the point of view of complexity, these are a *disastrous* ideas, but they hold the key to creating PCPs.

On paper the conversion is straight forward: define a witness of the **QUADEQ** instance (A, b) to be a map $f : \mathbb{F}_2^n \to \mathbb{F}_2$, given in terms of the ordered list of its values Γ_f, such that:

- Γ_f is the graph of a *linear* map $f : \mathbb{F}_2^n \to \mathbb{F}_2$

- its representing vector ($u_f \in \mathbb{F}_2^n$ such that $f = \langle u_f \mid - \rangle$) is a standard witness in the usual sense of **QUADEQ**, i.e. a vector satisfying $A\mathbf{U} = b$ where $\mathbf{U} = u_f \otimes u_f$.

While this makes sense, and produces precisely the same YES instances as the standard **QUADEQ**, $\widetilde{\mathbf{QUADEQ}}$ is not even in **NP**: one can't check linearity of the map represented by Γ_f without parsing all of its 2^n coordinates. If one drops the demand that Γ_f be the graph of a linear map, one can of course reconstruct *some* vector u'_f from Γ_f by reading the n bits representing the evaluations of f on the vectors $u'_1 = f({}^t(1,0,0,\ldots,0)), u'_2 = f({}^t(0,1,0,\ldots,0)), u'_3 = f({}^t(0,0,1,\ldots,0)) \ldots, u'_n = f({}^t(0,0,0,\ldots,1))$ and assembling the values in a vector $u'_f = (u'_1, u'_2, \ldots, u'_n)$. If f were linear, we'd have $u_f = u'_f$. Even so, this requires us to read at least n bits from Γ_f, and is therefore incompatible with the requirements of a PCP verifier to make a bounded number of requests to the proof.

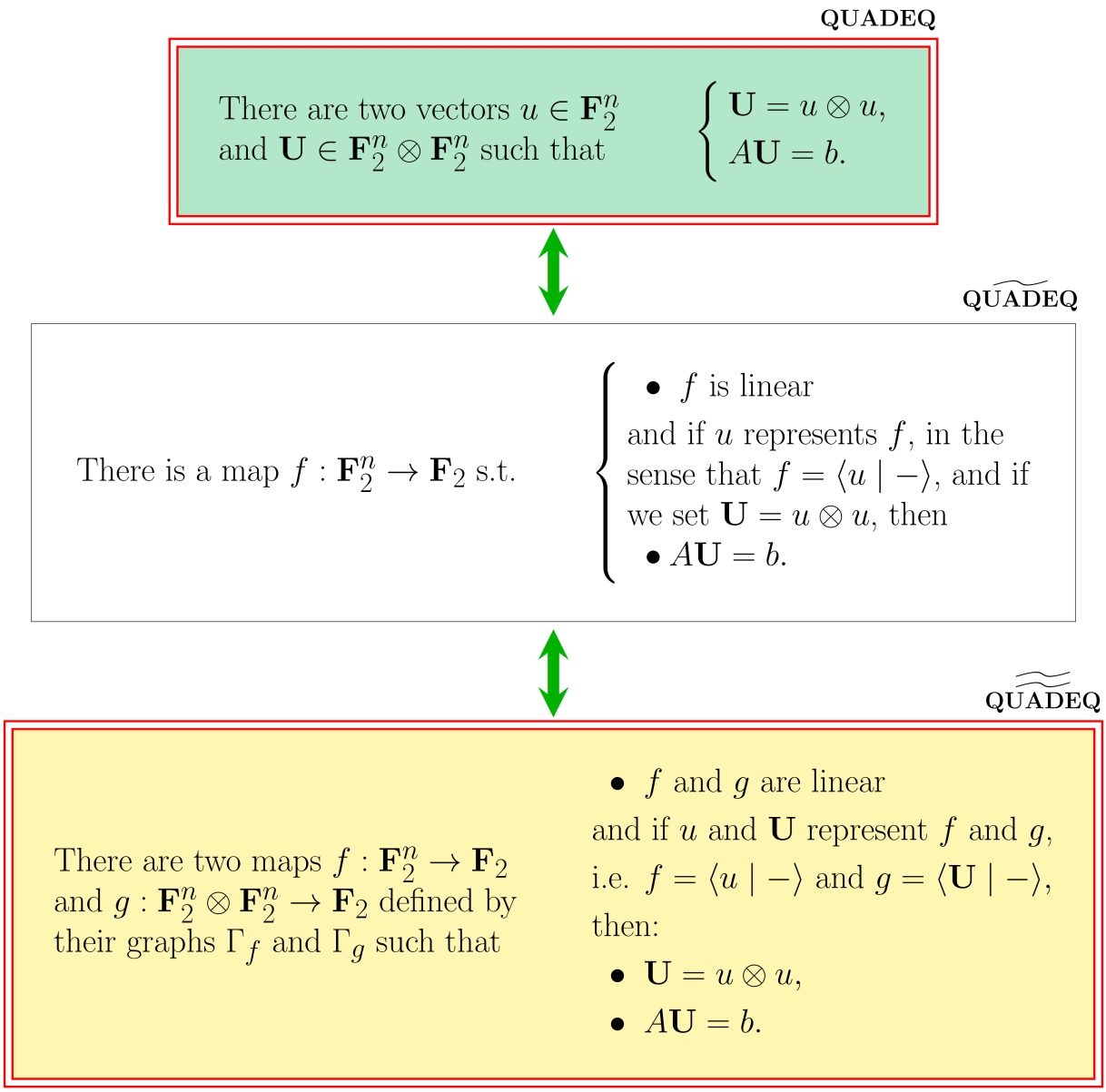
This proof format is truly bad: not only is it incredibly redundant (the encoding of vectors u as graphs $\Gamma_{\langle u|-\rangle}$ is), but it defines a witness only *implicitely*, as a vector u_f (supposedly) representing Γ_f satisfying $A(u_f \otimes u_f) = b$. It seems we've made things exponentially worse.

But we can make things worse, still. Why not have proofs be pairs of functions f, g, given in terms of their values Γ_f and Γ_g, where

- Γ_f is the graph of a *linear* map $f : \mathbb{F}_2^n \to \mathbb{F}_2$
- Γ_g is the graph of a *linear* map $g : \mathbb{F}_2^n \otimes \mathbb{F}_2^n \to \mathbb{F}_2$
- if u_f represents f (i.e. $u_f \in \mathbb{F}_2^n$ such that $f = \langle u_f \mid - \rangle$), and if $\mathbf{U}_g$ represents g (i.e. $\mathbf{U}_g \in \mathbb{F}_2^n \otimes \mathbb{F}_2^n$ such that $g = \langle \mathbf{U}_g \mid - \rangle$), then $\mathbf{U}_g = u_f \otimes u_f$ and $\mathbf{U}_g$ satisfies $A\mathbf{U}_g = b$.

Again, this colorful witness format (Γ_f, Γ_g) defines precisely the same YES instances as the standard proof format of **QUADEQ**, and compared to the previous variation on **QUADEQ**, it manages to be even more *implicitely defined* and more wasteful: now there are (supposedly) two implicitely defined vectors u_f and $\mathbf{U}_g$, a strong relationship between them $\mathbf{U}_g = u_f \otimes u_f$, satisfying $A\mathbf{U}_g = b$.

The following subsumes the transition from **QUADEQ** to $\widetilde{\widetilde{\mathbf{QUADEQ}}}$:



Collecting our thoughts

The "(Γ_f, Γ_g)" witness format of **QUADEQ** will be our new starting point, and this will be the PCP proof format for **QUADEQ**. The goal from now on is as follows: devise probabilistic tests for the three bullet points in the yellow box, and explain why these tests have the property that they rejects proofs for NO instances of **QUADEQ** with high probability, no matter the proof a dishonest prover may come up with.

The first test, "linearity tests" for Γ_f and Γ_g is the longest to set up. This is the purpose of the next section. Vector equatlity testing for the next two bullet points will be easier to set up.

Diving into the weeds: relative Hamming distance between maps and distance to linear maps

> From the NP-witness point of view, the transition from the **QUADEQ** to the **QU$\widetilde{\text{ADEQ}}$** witness format is worthless: they both convey the same information, but the latter is bloated with redundant derivatives of that information. Our goal here is not to improve on the NP witness format: our goal is to describe a PCP proof format. We will see how representing the NP witness u through (Γ_f, Γ_g) makes it amenable to probabilistic checking.

Our goal, as was just said, is to show that the "two functions (Γ_f, Γ_g)" provides a good format for **probabilistically checkable proofs**. If we take another look at what has to be checkedBut this sort of result doesn't just fall out of the sky. One needs a conceptual framework with which to interpret results and to tell us what to check for. The next few paragraphs lay down some vocabulary: Hamming distance between maps and ρ-quasi-linearity.

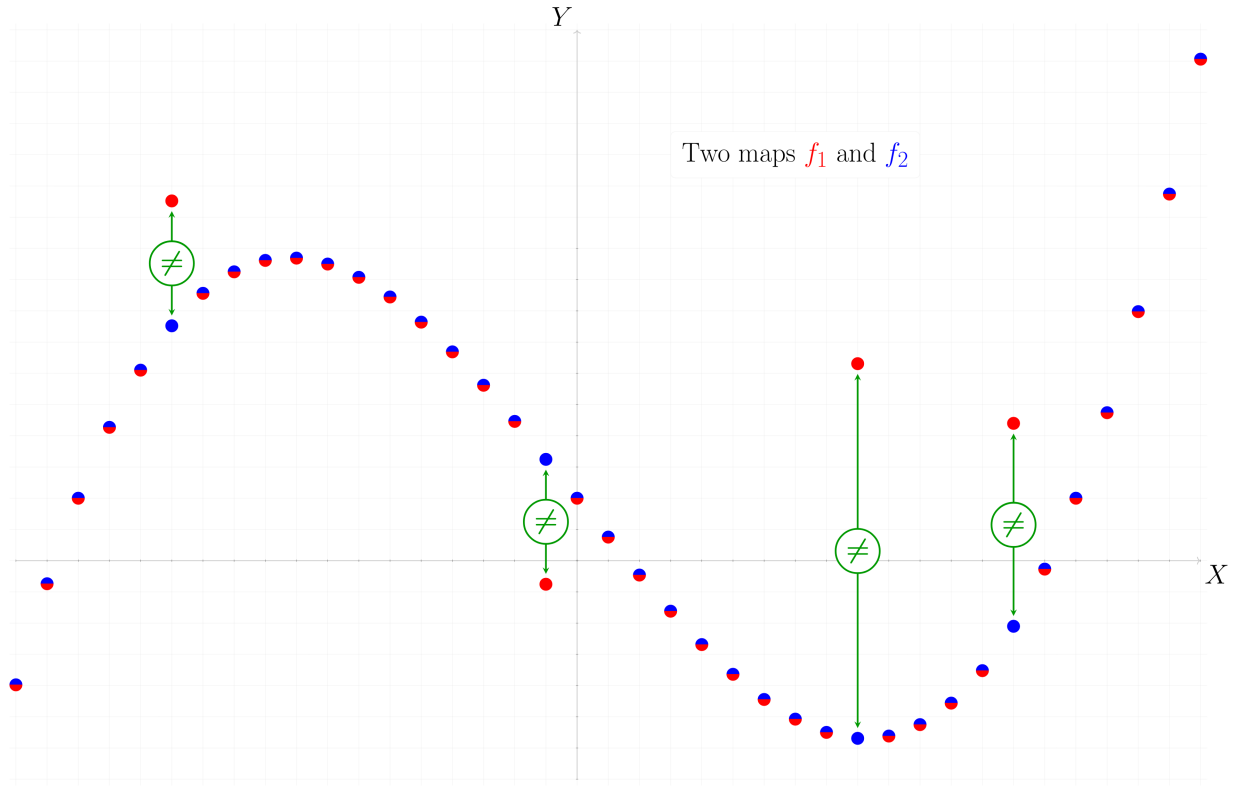
(Normalized) Hamming distance between maps

> The Hamming distance between two maps is the proportion of inputs on which they differ. Two maps are close in Hamming distance if they agree often.

Let's just briefly recall the **Hamming distance** between maps. Consider $f_1, f_2 : X \to Y$ two maps of sets between two finite sets X and Y. We may say that f_1 and f_2 are close to teach other if they agree on many of their inputs. Let's make this precise. Let $0 \leq \rho \leq 1$: f and g are ρ-close, or $\delta = 1 - \rho$ apart, if

$$\frac{\#\{x \in X \text{ s.t. } f_1(x) = f_2(x)\}}{\#X} \geq \rho \qquad \text{i.e.} \qquad \underbrace{\frac{\#\{x \in X \text{ s.t. } f_1(x) \neq f_2(x)\}}{\#X}}_{= d_{\mathsf{H}}(f_1, f_2)} \leq 1 - \rho$$

For instance the two following maps are close in that sense they only differ on 4 inputs (the values of the one are represented in red, while the values of the other are represented in blue) :

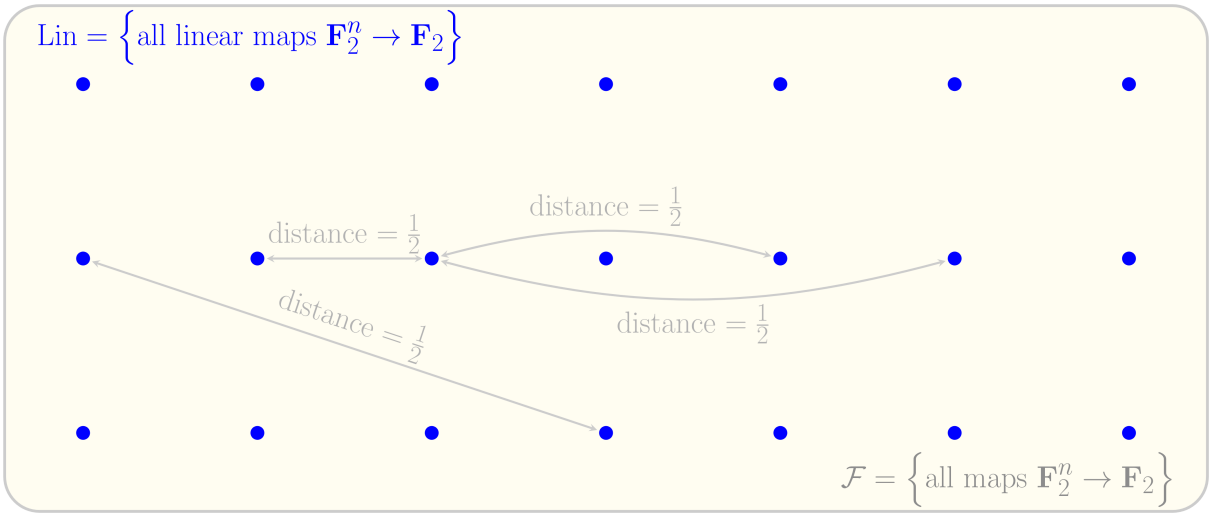


The number $d_{\mathsf{H}}(f_1, f_2) \in [0, 1]$ is the **relative Hamming distance** between the maps f_1 and f_2. The closer $d_{\mathsf{H}}(f_1, f_2)$ is to zero, the more f_1 and f_2 agree.

How the set of Linear functions sits inside the space of functions $\mathbb{F}_2^n \to \mathbb{F}_2$

> If use (relative) Hamming distance to measure distances in $\mathcal{F} = \{\mathbb{F}_2^n \to \mathbb{F}_2\}$, the space of *all* $\{0,1\}$-valued functions on $\mathbb{F}_2^n$, we see that the the set Lin of all *linear* functions occupies a very special configuration: any two distinct linear maps are precisely $\frac{1}{2}$ apart.

Here's how to picture how to picture Lin inside of $\mathcal{F}$: it's a discrete set of *equidistant* points:



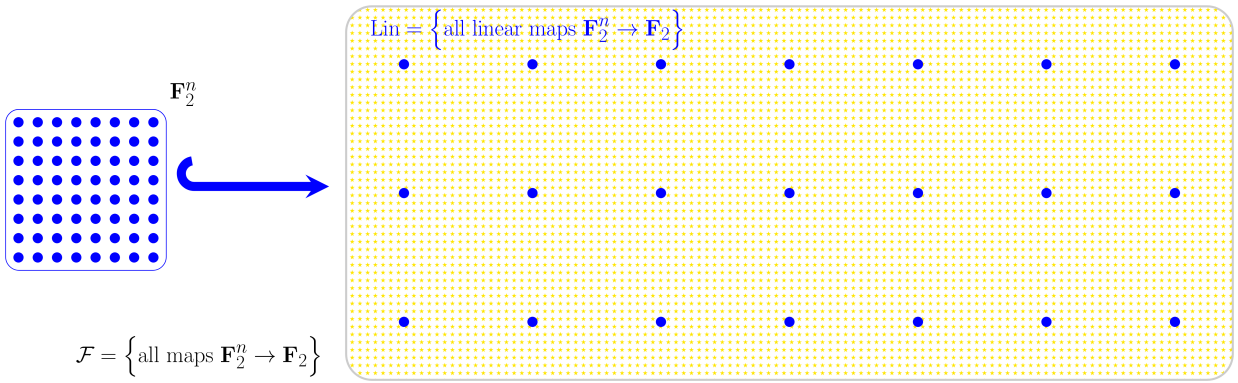
Indeed, in relative Hamming distance, two distinct linear maps $\alpha, \beta \in \mathrm{Lin}$ are *precisely* $\frac{1}{2}$-apart : $d_{\mathsf{H}}(\alpha, \beta) = \frac{1}{2}$. This follows from elementary linear algebra: if $\alpha \neq \beta$ are distinct linear maps, we get a nonzero linear map $\gamma = \beta - \alpha$ and a nontrivial partition of its domain into $\mathbb{F}_2^n = \{x \in \mathbb{F}_2^n \mid \gamma(x) = 0\} \sqcup \{x \in \mathbb{F}_2^n \mid \gamma(x) = 1\}$. Furthermore, it follows easily from linearity that if $\gamma(x_0) = 1$, then $\{x \in \mathbb{F}_2^n \mid \gamma(x) = 1\} = x_0 + \ker(\gamma)$, where $\ker(\gamma) = \{x \in \mathbb{F}_2^n \mid \gamma(x) = 0\}$. Thus the sets $\{x \in \mathbb{F}_2^n \mid \gamma(x) = 0\}$ and $\{x \in \mathbb{F}_2^n \mid \gamma(x) = 1\}$ are equinumerous, and so

$$d_{\mathsf{H}}(\alpha, \beta) = \frac{\#\{x \in \mathbb{F}_2^n \mid \gamma(x) = 1\}}{\#\big(\{x \in \mathbb{F}_2^n \mid \gamma(x) = 0\} \sqcup \{x \in \mathbb{F}_2^n \mid \gamma(x) = 1\}\big)} = \frac{1}{2}.$$

Uniform Separation

> Since *linear* functions $\mathbb{F}_2^n \to \mathbb{F}_2$ are in one-to-one correspondence with vectors $\mathbb{F}_2^n$, this picture tells us how the vectors $\mathbb{F}_2^n$ embed into $\mathcal{F}$. While vectors may be uncomfortably close in $\mathbb{F}_2^n$ (all distances in $\{0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n-1}{n}, 1\}$ are possible), they are nicely separated when viewed as (linear) functions $\langle v \mid - \rangle$ in $\mathcal{F}$.

If we have two distinct vectors $u = (u_1, u_2, \ldots, u_n)$ and $v = (v_1, v_2, \ldots, v_n)$ in, $\mathbb{F}_2^n$, their relative Hamming distance is $d_{\mathsf{H}}(u, v) =$ (number of indices i such that $u_i \neq v_i)/n$ can be anything in $\{\frac{1}{n}, \frac{2}{n}, \ldots, \frac{n-1}{n}, 1\}$. However, the relative Hamming distance between the two associated linear maps $\langle u \mid - \rangle$ and $\langle v \mid - \rangle$ as measured inside of $(\mathcal{F}, d_{\mathsf{H}})$ is necessarily $\frac{1}{2}$.



By embedding vectors into the function space $\mathcal{F}$ we have created **uniform separation**. Note that in the picture above, the yellow stars represent individual functions, i.e. points in $\mathcal{F}$, and the blue dots represent the linear functions. The following subsumes this discussion:

$$\underbrace{u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \in \mathbf{F}_2^n}_{\text{described by } \boldsymbol{n} \text{ bits}} \quad \rightsquigarrow \quad \underbrace{\langle u \mid - \rangle : \begin{cases} \mathbf{F}_2^n & \longrightarrow \mathbf{F}_2 \\ (x_1, \ldots, x_n) & \longmapsto \sum_{i=1}^n x_i u_i \end{cases}}_{\text{described by } \mathbf{2^n} \text{ bits}}$$

$$\left\{ \begin{array}{c} u \neq v \text{ may be as close as } \frac{1}{n} \\ \text{in rel. Hamming distance} \end{array} \right\} \quad \rightsquigarrow \quad \left\{ \begin{array}{c} \langle u \mid - \rangle \neq \langle v \mid - \rangle \text{ are } \mathbf{always} \ \frac{1}{2} \\ \text{apart in rel. Hamming distance} \end{array} \right\}$$

Hamming distance of a map to the set of linear maps

> Establishing linearity of a map $f : \mathbb{F}_2^n \to \mathbb{F}_2$ given by its values Γ_f requires reading all of f's values. Checking that that map agrees with *some specified linear map* φ on a large proportion of all inputs can be done probabilistically: pick random values x and compare $f(x)$ and $\varphi(x)$, and repeat a certain amount of times. We introduce the set of maps that agree with *some linear map* on a large proportion of inputs.

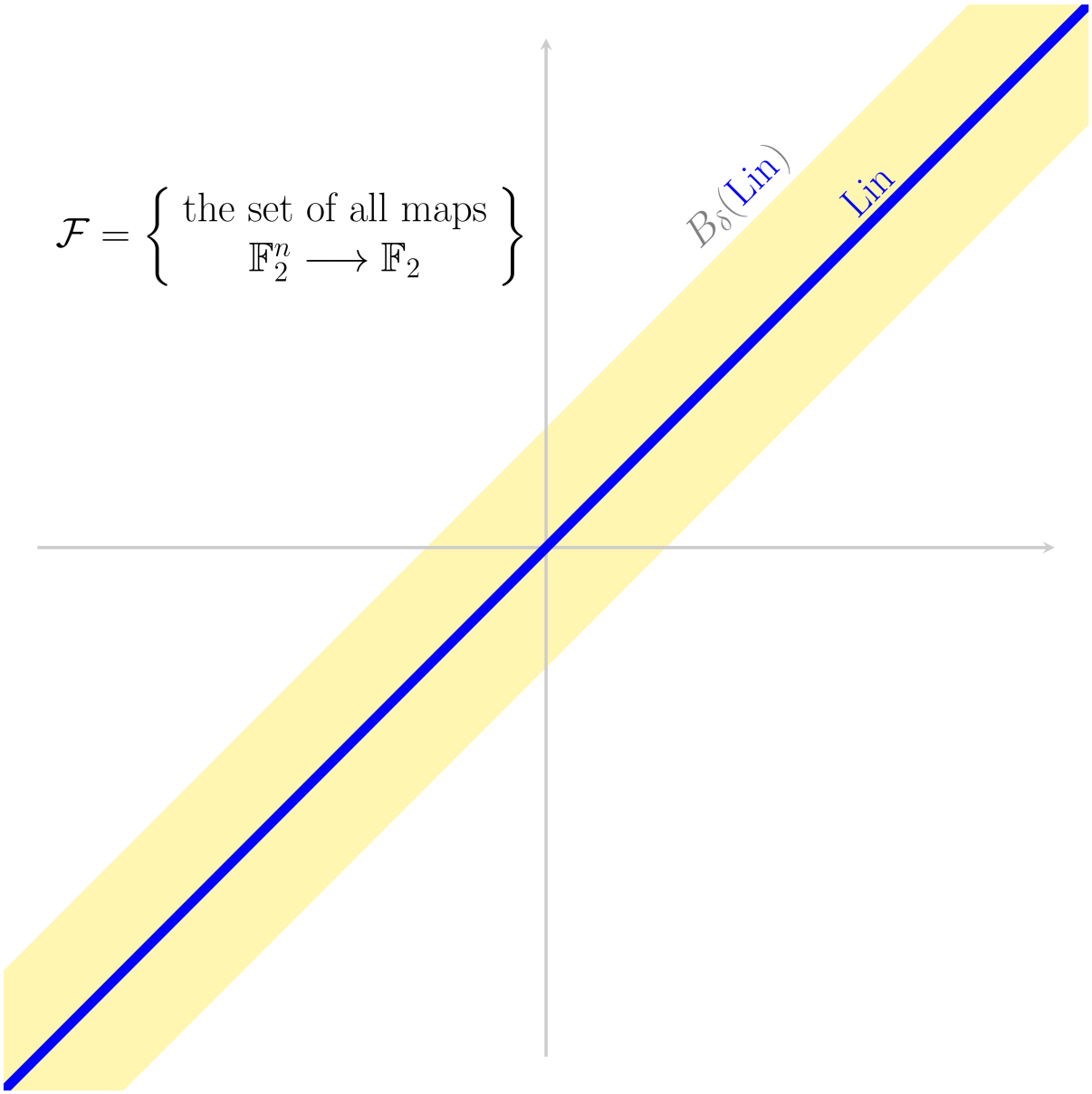
Using relative Hamming distance, we can define what it means for a given map f to be $1 - \delta = \rho$-close to *some specified* linear map φ : $d_{\mathsf{H}}(f, \varphi) \leq \delta$. From there we can define what it means for a map to be ρ-close to *the set* of all linear maps: in order to be ρ-close to the set of linear maps $\mathrm{Lin}_{\mathbb{F}_2}(\mathbb{F}_2^n, \mathbb{F}_2)$ we ask that there exist *som* linear map $\varphi : \mathbb{F}_2^n \to \mathbb{F}_2$ with $d_{\mathsf{H}}(f, \varphi) \leq 1 - \rho$. Let us write $B_\rho(\mathrm{Lin})$ the set of those maps that are ρ-close to *some* linear map $\mathbb{F}_2^n \to \mathbb{F}_2$. In other words:

$$B_\delta(\mathrm{Lin}) = \{f \in \mathcal{F} \mid \exists \varphi \in \mathrm{Lin},\ d_{\mathsf{H}}(f, \varphi) \leq \delta\} = \bigcup_{\varphi \in \mathrm{Lin}} B_\delta(\varphi)$$

where $B_\delta(\varphi)$ stands for the set of all maps f with $d_{\mathsf{H}}(f, \varphi) \leq \delta$. For instance, if we take $\rho = 0.01$, $B_\rho(\mathrm{Lin})$ is the collection of all maps $f : \mathbb{F}_2^n, \mathbb{F}_2$ such that there exists some linear map $\varphi : \mathbb{F}_2^n, \mathbb{F}_2$ such that f and φ agree on at least 99% of all inputs. I.e. f is "almost linear" in some sense.

Of course we can make similar definitions for maps $g_1, g_2 : \mathbb{F}_2^n \otimes \mathbb{F}_2^n \to \mathbb{F}_2$ and linear maps $\mathrm{Lin}_{\mathbb{F}_2}(\mathbb{F}_2^n \otimes \mathbb{F}_2^n, \mathbb{F}_2)$.

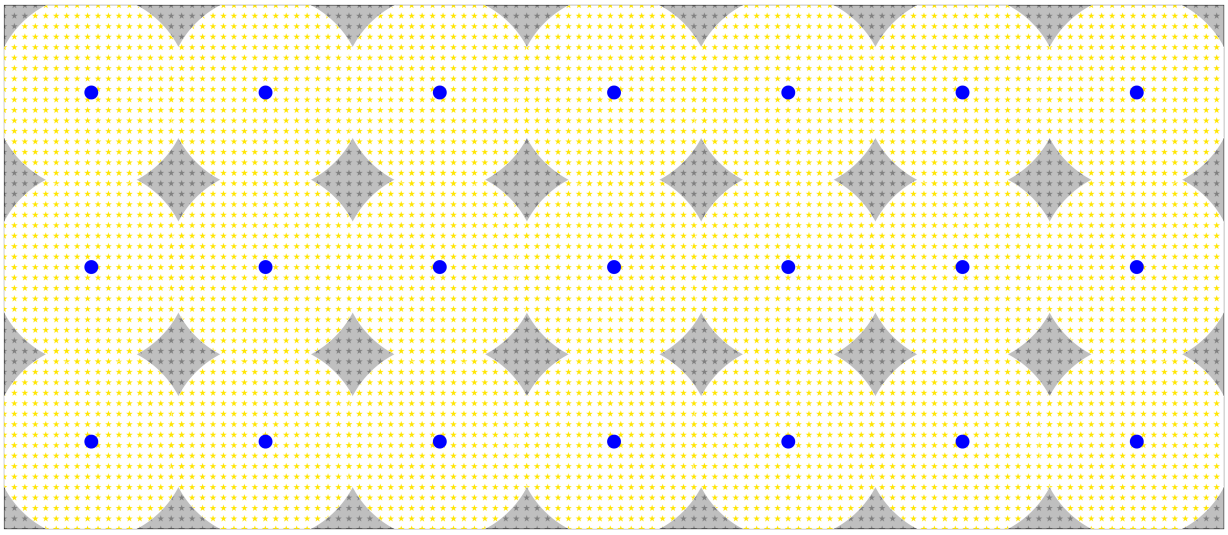
One can imagine the set of linear maps $\mathrm{Lin} = \mathrm{Lin}_{\mathbb{F}_2}(\mathbb{F}_2^n, \mathbb{F}_2)$ sitting in the set $\mathcal{F}$ of all maps $\mathbb{F}_2^n \to \mathbb{F}_2$ as a very thin "linear subspace", and $B_\rho(\mathrm{Lin})$ being a sort of thickening of that linear subspace.



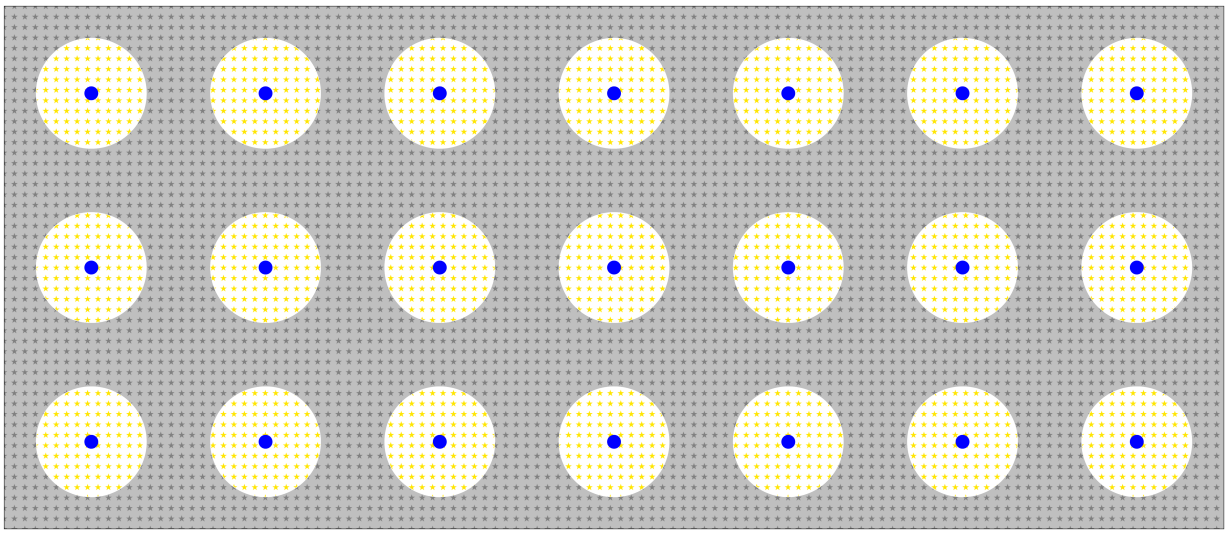
What $B_\rho(\text{Lin})$ looks like for different values of ρ

> If ρ is small, the balls that make up $B_\rho(\text{Lin})$ are disjoint, and any function in $B_\rho(\text{Lin})$ has a single closest linear neighbor.

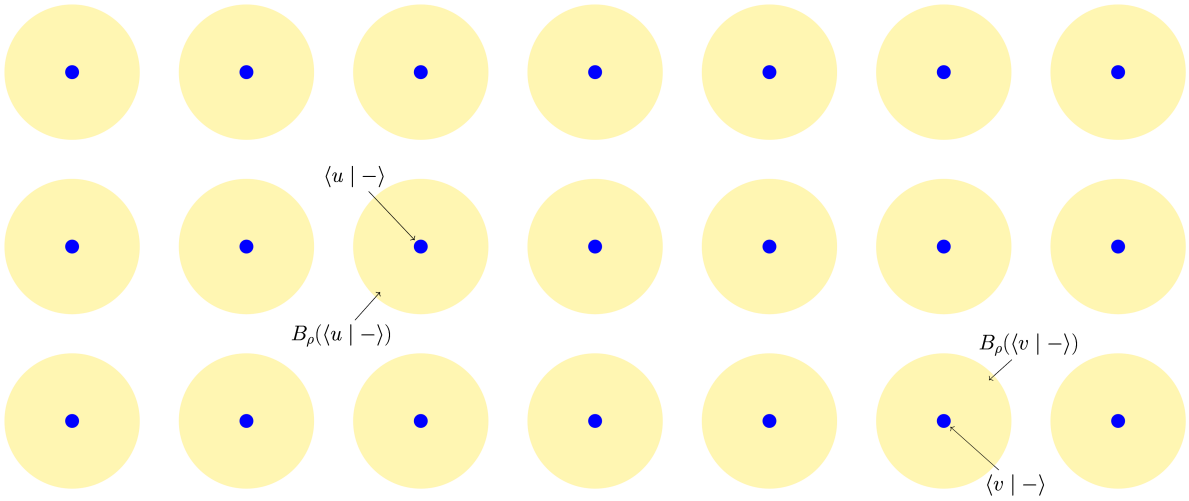
In both pictures below, the blue dots represent the linear functions (i.e. the points of Lin) as they sit in the space $\mathcal{F}$ of all functions $\mathbb{F}_2^n \to \mathbb{F}_2$. Stars represent other functions. The gray stars in the shaded regions represent the functions which lie outside of the neighborhood $B_\rho(\text{Lin})$. The yellow stars on the other hand represent individual functions in the ρ-neighborhood of Lin. When $\rho \geq \frac{1}{4}$ the individual balls $B_\rho(\varphi)$, for $\varphi \in \text{Lin}$ which collectively make up the ρ-neighborhood of Lin overlap:



In other words, when $\rho \geq \frac{1}{4}$, a map $f \in B_\rho(\mathrm{Lin})$ might have multiple "closest" linear functions. But when $\rho < \frac{1}{4}$ the balls are disjoint. Indeed, if $\rho < \frac{1}{4}$ and φ, ψ are two distinct linear maps, then the balls $B_\rho(\varphi) = \{\text{all maps f with } d_{\mathsf{H}}(f, \varphi) < \rho\}$ and $B_\rho(\psi) = \{\text{all maps f with } d_{\mathsf{H}}(f, \psi) < \rho\}$ are disjoint. For if there was a map f with $B_\rho(\varphi) \cap B_\rho(\psi)$, then we would find $\frac{1}{2} = d_{\mathsf{H}}(\varphi, \psi) \leq d_{\mathsf{H}}(\varphi, f) + d_{\mathsf{H}}(f, \psi) < 2\rho < \frac{1}{2}$, which is absurd. Thus for $\rho < \frac{1}{4}$, the picture of $B_\rho(\mathrm{Lin})$ is as follows:



If we remove from the picture the maps that don't lie in the neighbourhood $B_\rho(\mathrm{Lin})$, and neglect to draw individual points representing fuctions and switch to "solid balls", the picture for $B_\rho(\mathrm{Lin})$ with $\delta < \frac{1}{4}$ is as follows:



One might hope that a map $f \in B_\rho(\text{Lin})$, for $\rho < \frac{1}{4}$, should in some sense, ''remember" the unique linear map closest to it, and we can hope that there should be a way to extract that unique linear map φ with $f \in B_\rho(\varphi)$ from f.

ρ-Quasi linearity

> Linearity is fragile : change one bit in a linear map and you lose linearity. Therefore we can only hope for a test for some form of "approximate linearity". We already saw one notion of "approximate linearity": a map could be said to be approximatively linear if it agrees of some high proportion $\rho = 1 - \delta > 1 - \frac{1}{4}$ of all its inputs with a given linear map, in other words, if $f \in B_\delta(\text{Lin})$. As was discussed in the previous paragraph, such a map then has a *unique* linear map in its δ-vicinity.
>
> However, unless we know in advance which linear map φ supposedly contains f in its δ neighborhood, it seems we can't test agreement of f that linear map.
>
> In this paragraph we introduce ρ-Quasi linearity, an alternate notion of "approximate linearity", and show how the two notions interact. The great advantage this notion of approximate linearity has over the previous one is that it is self-contained: one does not have to refer to some other linear map one does not really know.

Recall that a map $\varphi : \mathbb{F}_2^n \to \mathbb{F}_2$ is **linear** if for all $x, y \in \mathbb{F}_2^p$, $\varphi(x+y) = \varphi(x) + \varphi(y)$. Let $\rho = 1 - \delta \in]0, 1[$. We shall say that a map $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is **ρ-quasi linear** if

$$\frac{\#\left\{(x,y) \in \mathbf{F}_2^n \times \mathbf{F}_2^n \text{ s.t. } f(x) + f(y) = f(x+y)\right\}}{\#(\mathbf{F}_2^n \times \mathbf{F}_2^n)} \geq \rho$$

Approximate linearity is a kind of *structural* proximity to linear maps. Note that this structural proximity to linear maps is inherently quite different from the Hamming proximity :

- if we want to test for *Hamming proximity* to some linear map ... well we already need to have a candidate linear map against which to test agreement,
- on the other hand, *structural proximity* to linear maps is defined purely in terms of the map itself, i.e. it can be tested with knowledge of f alone.

The following very nice result is thus quite remarkable :

Theorem 1. *If f is ρ-quasi linear, with $\rho > \frac{1}{2}$, then $f \in B_\rho(\mathrm{Lin})$, i.e. there is a linear map φ such that f is ρ-proximate to φ.*

Here is a proof. It will be convenient to work with maps $f : \mathbb{F}_2^p \to \{-1, +1\} \subset \mathbb{R}$ rather than the usual $f : \mathbb{F}_2^p \to \mathbb{F}_2 = \{0, 1\}$. It is easy to convert the former to the latter, for example we can set $\widehat{f} = (-1)^f$. We can endow the set of all real-valued maps $\mathbb{F}_2^p \to \mathbb{R}$ on $\mathbb{F}_2^p$ with an inner product given by

$$\langle \widehat{f} \mid \widehat{g} \rangle := \frac{1}{2^p} \sum_{v \in \mathbb{F}_2^p} \widehat{f}(v)\widehat{g}(v)$$

One notices that for any functions $f, g \in \mathcal{F}$ (i.e. maps $\mathbb{F}_2^p \to \mathbb{F}_2$), one has

$$\begin{aligned}
\langle \widehat{f} \mid \widehat{g} \rangle &= \frac{1}{2^p}\Big(\sum_{\substack{v \in \mathbb{F}_2^p \\ f(v)=g(v)}} (-1)^{f(v)}(-1)^{g(v)} + \sum_{\substack{v \in \mathbb{F}_2^p \\ f(v)\neq g(v)}} (-1)^{f(v)}(-1)^{g(v)} \Big) \\
&= \frac{1}{2^p}\Big(\sum_{\substack{v \in \mathbb{F}_2^p \\ f(v)=g(v)}} 1 - \sum_{\substack{v \in \mathbb{F}_2^p \\ f(v)\neq g(v)}} 1 \Big) \\
&= \frac{1}{2^p}\Big(\Big[2^p - \#\{v \in \mathbb{F}_2^p \mid f(v) \neq g(v)\}\Big] - \#\{v \in \mathbb{F}_2^p \mid f(v) \neq g(v)\} \Big) \\
&= 1 - 2d_{\mathsf{H}}(f, g)
\end{aligned}$$

Let us write $\chi_v = \widehat{\langle v \mid - \rangle}$ for simplicity. If follows that for any $v, w \in \mathbb{F}_2^p$

$$\langle \chi_v \mid \chi_w \rangle = \begin{cases} \text{if } v = w : & 1 - 2 \cdot 0 = 1 \\ \text{if } v \neq w : & 1 - 2 \cdot \frac{1}{2} = 0 \end{cases}$$

The maps $\big(\chi_v\big)_{v \in \mathbb{F}_2^p}$ thus form an orthonormal family, and indeed an orthonormal basis of $\widehat{\mathcal{F}} = (\{\text{all maps } \mathbb{F}_2^p \to \mathbb{R}\}, \langle - \mid - \rangle)$. Any function real valued function $f : \mathbb{F}_2^p \to \mathbb{R}$ thus decomposes as

$$f = \sum_{v \in \mathbb{F}_2^p} \langle \chi_v \mid f \rangle \cdot \chi_v$$

Now if $f : \mathbb{F}_2^p \to \mathbb{F}_2$, let us consider the two maps $A : \mathbb{F}_2^p \times \mathbb{F}_2^p \to \mathbb{F}_2$ and $B : \mathbb{F}_2^p \times \mathbb{F}_2^p \to \mathbb{F}_2$ defined by the formulas

$$A(x, y) := f(x + y) \quad \text{and} \quad B(x, y) := f(x) + f(y)$$

The decompositions of A and B relative to the orthonormal basis $(\chi_{v \oplus w})_{(v,w)\mathbb{F}_2^p \times \mathbb{F}_2^p}$ follows from that of f, indeed one has, for all $x, y \in \mathbb{F}_2^p \times \mathbb{F}_2^p$,

$$\begin{aligned}
\widehat{A}(x, y) = \widehat{f}(x + y) &= \sum_{u \in \mathbb{F}_2^p} \langle \chi_u \mid \widehat{f} \rangle \cdot \overbrace{\chi_u(x + y)}^{\chi_u(x)\chi_u(y)} \\
&= \sum_{u \in \mathbb{F}_2^p} \langle \chi_u \mid \widehat{f} \rangle \cdot \chi_{u \oplus u}(x \oplus y)
\end{aligned}$$

so that and

$$\widehat{A} = \sum_{u \in \mathbb{F}_2^p} \langle \chi_u \mid \widehat{f} \rangle \chi_{u \oplus u}$$

while for all $x, y \in \mathbb{F}_2^p \times \mathbb{F}_2^p$,

$$\begin{aligned}
\widehat{B}(x, y) = \widehat{f}(x)\widehat{f}(y) &= \Big(\sum_{v \in \mathbb{F}_2^p} \langle \chi_v \mid \widehat{f} \rangle \chi_v(x) \Big) \cdot \Big(\sum_{w \in \mathbb{F}_2^p} \langle \chi_w \mid \widehat{f} \rangle \chi_w(y) \Big) \\
&= \sum_{v, w \in \mathbb{F}_2^p} \langle \chi_v \mid \widehat{f} \rangle \langle \chi_w \mid \widehat{f} \rangle \cdot \underbrace{\chi_v(x)\chi_w(y)}_{= \chi_{v \oplus w}(x \oplus y)}
\end{aligned}$$

so that

$$\widehat{B} = \sum_{v,w\in\mathbb{F}_2^p} \langle \chi_v \mid \widehat{f}\rangle\langle \chi_w \mid \widehat{f}\rangle \cdot \chi_{v\oplus w}$$

Then

$$\begin{aligned}
1 - 2d_{\mathsf{H}}(A,B) = \langle \widehat{A} \mid \widehat{B}\rangle &= \left\langle \sum_{u\in\mathbb{F}_2^p} \langle \chi_u \mid \widehat{f}\rangle \chi_{u\oplus u} \;\middle|\; \sum_{v,w\in\mathbb{F}_2^p} \langle \chi_v \mid \widehat{f}\rangle\langle \chi_w \mid \widehat{f}\rangle \cdot \chi_{v\oplus w} \right\rangle \\
&= \sum_{u,v,w\in\mathbb{F}_2^p} \langle \chi_u \mid \widehat{f}\rangle\langle \chi_v \mid \widehat{f}\rangle\langle \chi_w \mid \widehat{f}\rangle \cdot \langle \chi_{u\oplus u} \mid \chi_{v\oplus w}\rangle \\
&\overset{(\star)}{=} \sum_{u\in\mathbb{F}_2^p} \langle \chi_u \mid \widehat{f}\rangle^3 \\
&\leq \underbrace{\Big(\sum_{u\in\mathbb{F}_2^p} \langle \chi_u \mid \widehat{f}\rangle^2\Big)}_{\overset{(\star\star)}{=}1} \max_u\{\langle \chi_u \mid \widehat{f}\rangle\} \\
&= \max_u\{1 - 2d_{\mathsf{H}}(f, \langle u \mid -\rangle)\} \\
&= 1 - 2\min_u d_{\mathsf{H}}(f, \langle u \mid -\rangle) \\
&= 1 - 2d_{\mathsf{H}}(f, \mathrm{Lin})
\end{aligned}$$

and so $d_{\mathsf{H}}(f,\mathrm{Lin}) \leq d_{\mathsf{H}}(A,B)$. Equality $(\star)$ follows from the orthogonality of the χ_v : if $u = v = w$, then $\langle \chi_{u\oplus u} \mid \chi_{v\oplus w}\rangle = 1$, in all other cases, $\langle \chi_{u\oplus u} \mid \chi_{v\oplus w}\rangle = 0$ equality $(\star\star)$ follows from the fact that $\sum_{u\in\mathbb{F}_2^p}\langle \chi_u \mid \widehat{f}\rangle^2 = \langle f \mid f\rangle = 1 - 2d_{\mathsf{H}}(f,f) = 1$. Looking at the definition of $d_{\mathsf{H}}(A,B)$, we see that

$$d_{\mathsf{H}}(A,B) = \frac{\#\{(x,y)\in\mathbb{F}_2^p\times\mathbb{F}_2^p \mid f(x+y)\neq f(x)+f(y)\}}{2^p\times 2^p}.$$

Now let us assume f is $1-\delta=\rho$-quasi linear. Then if $u_0\in\mathbb{F}_2^p$ realizes the minimum $d_{\mathsf{H}}(f,\mathrm{Lin}) = d_{\mathsf{H}}(f,\langle u_0 \mid -\rangle)$, then

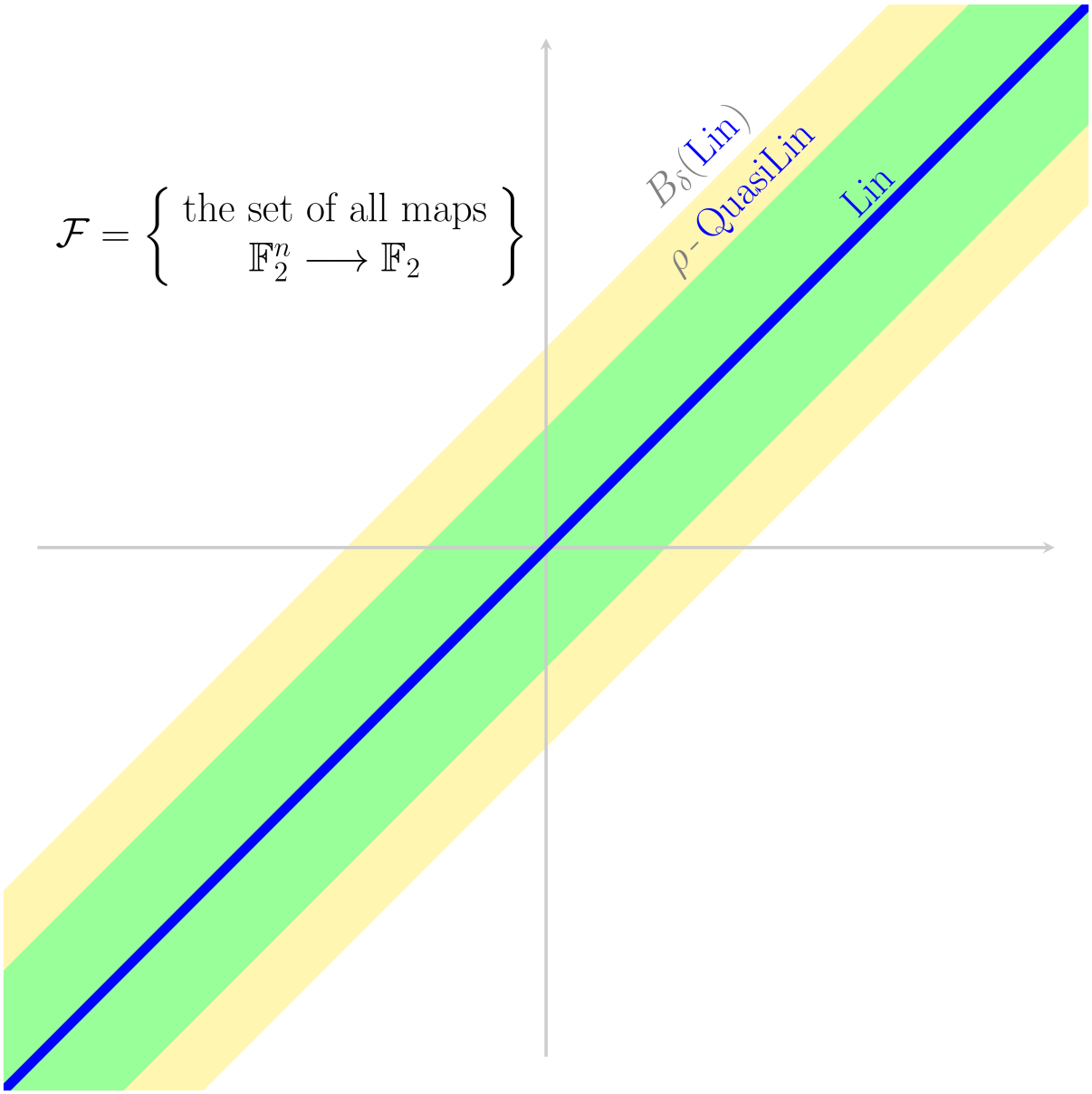
$$d_{\mathsf{H}}(f,\langle u_0 \mid -\rangle) = d_{\mathsf{H}}(f,\mathrm{Lin}) \leq d_{\mathsf{H}}(A,B) \leq \delta$$

and so $f\in B_\delta(\mathrm{Lin})$.

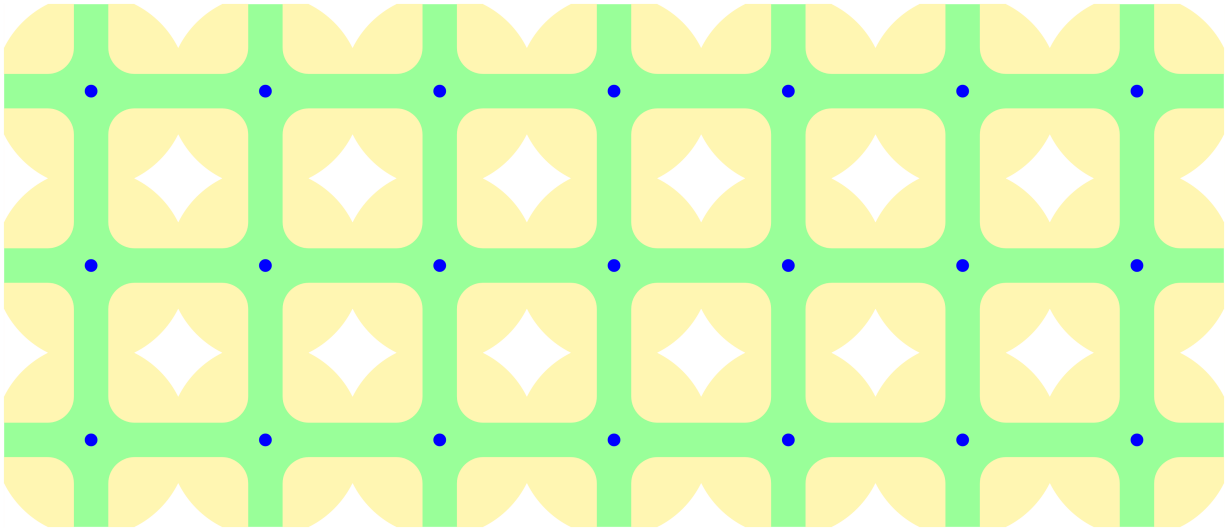
How to picture the set of ρ-quasi-linear maps within $B_\delta(\mathrm{Lin})$

> Here are a few pictures giving visual intuition of how the set of ρ-quasi-linear map sits within the δ-neighborhood $B_\delta(\mathrm{Lin})$ of all linear maps and within the set $\mathcal{F}$ of all maps $\mathbb{F}_2^n\to\mathbb{F}_2$.

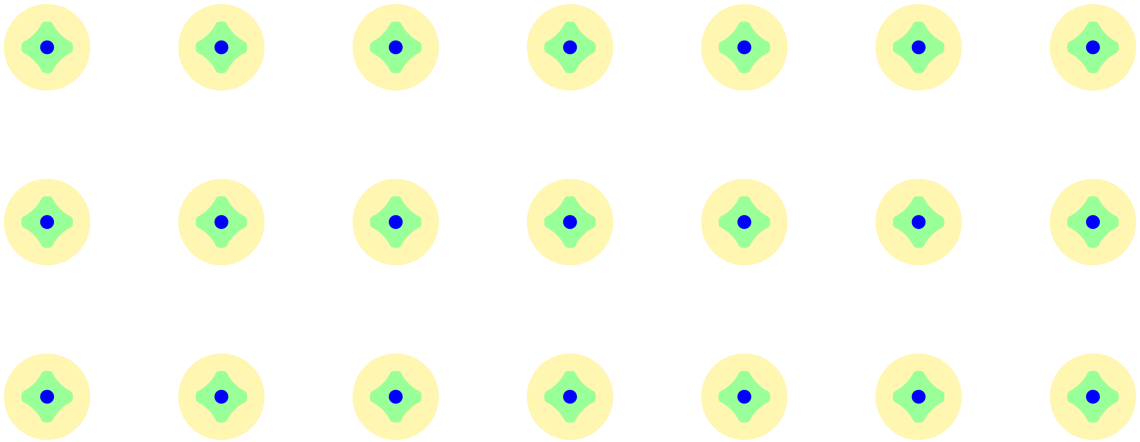
The content of the previous theorem is that ρ-quasi-linear maps are automatically in $B_\delta(\mathrm{Lin})$, where $\rho = 1-\delta$.



We can picture this inclusion for $\delta \geq \frac{1}{4}$: overlapping balls $B_\delta(\varphi)$ (one for each linear map $\varphi \in \text{Lin}$) filled in with yellow, the totality of these maps containing all ρ-quasi-linear maps, represented as the green subset:



and $\delta < \frac{1}{4}$, we again have neatly separated balls $B_\delta(\varphi)$ for all linear maps $\varphi \in \mathrm{Lin}$ and filled in with yellow, each containing a subset (in green) of all ρ-quasi-linear maps:



Test #1 : ρ-quasi-linearity test for functions

> The great improvement the notion of ρ-quasi-linearity brings over proximity in Hamming distance to the set of linear maps (i.e. to *some* linear map) is that this definition is *self-contained*. To (probabilistically) test for ρ-quasi-linearity of a map $f : \mathbb{F}_2^n \to \mathbb{F}_2$, one only needs to have access to the values of f, which is precisely what the "(Γ_f, Γ_g)" PCP proof format provides.
>
> We describe a test that takes as input a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ given by its values Γ_f and with probability $> 1/2$ rejects if it isn't at least ρ-quasi-linear, while accepting linear functions 100% of the time.

Let $0 < \rho < 1$, and write $\rho = 1 - \delta$. Let us put $\mathsf{R} = \mathsf{R}_\delta = \lceil \frac{1}{\delta} \rceil$.

ρ-quasi-linearityTEST $\mathcal{T}_\delta$

1. pick two points x and y in $\mathbb{F}_2^n$ uniformly at random
2. query Γ_f at x, y and $x+y$,
3. compare $f(x+y)$ and $f(x)+f(y)$; `Reject` if they disagree;
4. Repeat R many times and `Accept`

Clearly, if f is a linear function, it passes the test with flying colours. The interesting question is: what happens if f isn't ρ-quasi-linear? Suppose $f : \mathbb{F}_2^n \to \mathbb{F}_2$ fails to be ρ-quasi-linear. Let us set

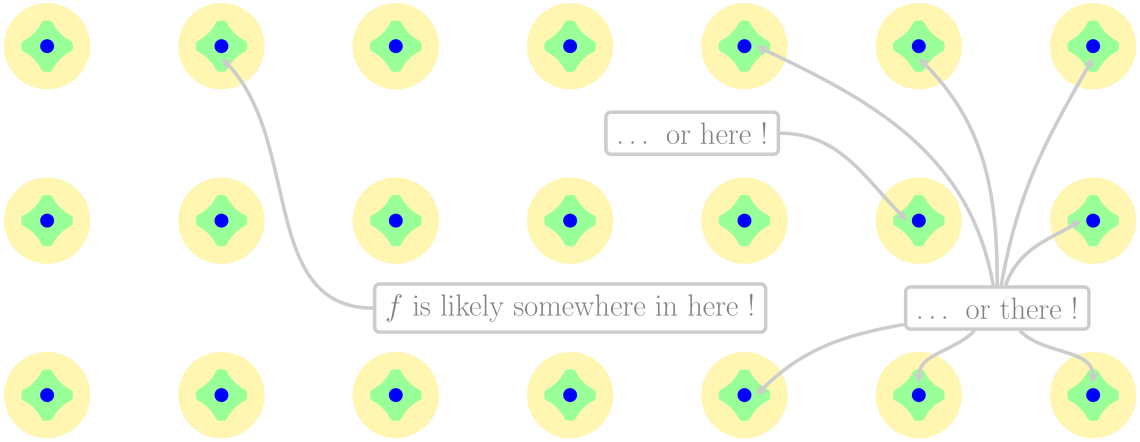
$$\frac{\#\left\{(x,y) \in \mathbf{F}_2^n \times \mathbf{F}_2^n \text{ s.t. } f(x)+f(y)=f(x+y)\right\}}{\#(\mathbf{F}_2^n \times \mathbf{F}_2^n)} = \underbrace{\rho_f}_{=1-\delta_f} < \underbrace{\rho}_{=1-\delta}$$

We can compute the probability of f passing the test $\mathcal{T}_\delta$. We find that it is $< \frac{1}{2}$, indeed :

$$\begin{aligned}
\mathbb{P}\Big[\mathcal{T}_\delta(f) = \mathtt{Accept}\Big] &= \rho_f^{\mathsf{R}} \\
&= \big(1-\delta_f\big)^{\lceil \frac{1}{\delta} \rceil} \\
&\leq \big(1-\delta\big)^{\frac{1}{\delta}} \\
&< \frac{1}{e} \\
&< \frac{1}{2}
\end{aligned}$$

(This follows from the concavity of the natural log).

Ok, so what can you conclude from running this test on a function f? Well if f isn't ρ-quasi-linear, then the probability of f passing, say, twenty iterations of $\mathcal{T}_\delta$ is less than one in 485 Million, and likely f would have been rejected. It is thus likely (and we will make this precise at the very end) that f is in one of the green regions depicted below, since the complement of the green regions reprensents the collection of all maps which aren't ρ-quasi-linear.

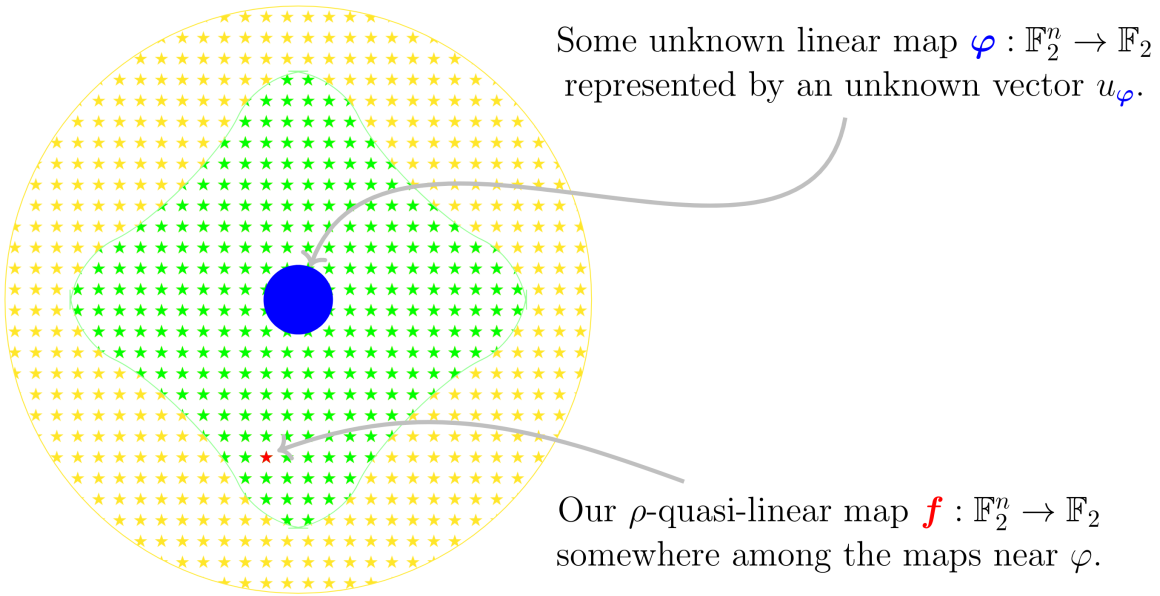


The map f is therefore likely to be structurally very close to being linear.

Focus change

> The two tests that follow are about (probabilistically) establishing equalities between vectors. That on its own sounds like it should be simple enough. However, the vectors in question are only **implicitely defined**.

Suppose we are given a function f that is ρ-quasi-linear for $\rho = 1 - \delta$ close to 1. We know from — **REFERENCE PLEASE** — that there is some linear map φ with $d_{\mathsf{H}}(f, \varphi) \leq \delta$. We can picture that function f as lying in the immediate vicinity of some **unknown** linear map $\varphi = \langle u_\varphi \mid - \rangle$ like so:



Again, we use yellow stars to represent functions in the ball $B_\delta(\varphi)$, and green stars to represent ρ-quasi-linear maps in that neighborhood. Our map f is one of those green points (but we chose to highlight it in red).

In the case at hand, we will have two such ρ-quasi linear maps $f : \mathbb{F}_2^n \to \mathbb{F}_2$ and $g : \mathbb{F}_2^n \otimes \mathbb{F}_2^n \to \mathbb{F}_2$ given by their values Γ_f and Γ_g. The linear maps representing their closest linear neighbour are $\varphi = \langle u \mid - \rangle$ (for f) and $\psi = \langle \mathbf{U} \mid - \rangle$ (for g), with $u \in \mathbb{F}_2^n$ and $\mathbf{U} \in \mathbb{F}_2^n \otimes \mathbb{F}_2^n$. But what we are interested in is probing properties of the **implicitely defined vectors** u and $\mathbf{U}$, namely whether:

$$\mathbf{U} \stackrel{?}{=} u \otimes u \quad \text{and} \quad A\mathbf{U} \stackrel{?}{=} b$$

But we can't access φ or ψ directly: the only tool in our toolbox is querying f and g.

The upsides of working with well-behaved objects

Ideally we would work directly with the vectors $u \in \mathbb{F}_2^n$ and $\mathbf{U} \in \mathbb{F}_2^n \otimes \mathbb{F}_2^n$, or equivalently with the linear map $\varphi : x \mapsto \langle u \mid x \rangle$ and $\psi : \mathbf{X} \mapsto \langle \mathbf{U} \mid \mathbf{X} \rangle$. By virtue of the identity we're after ($\mathbf{U} \stackrel{?}{=} u \otimes u$), it is convenient to try to work with the **bilinear map** associated with $\mathbf{U}$:

$$\begin{array}{rcl} \mathbb{F}_2^n \times \mathbb{F}_2^n & \longrightarrow & \mathbb{F}_2 \\ (x, y) & \longmapsto & \langle \mathbf{U} \mid x \otimes y \rangle \end{array}$$

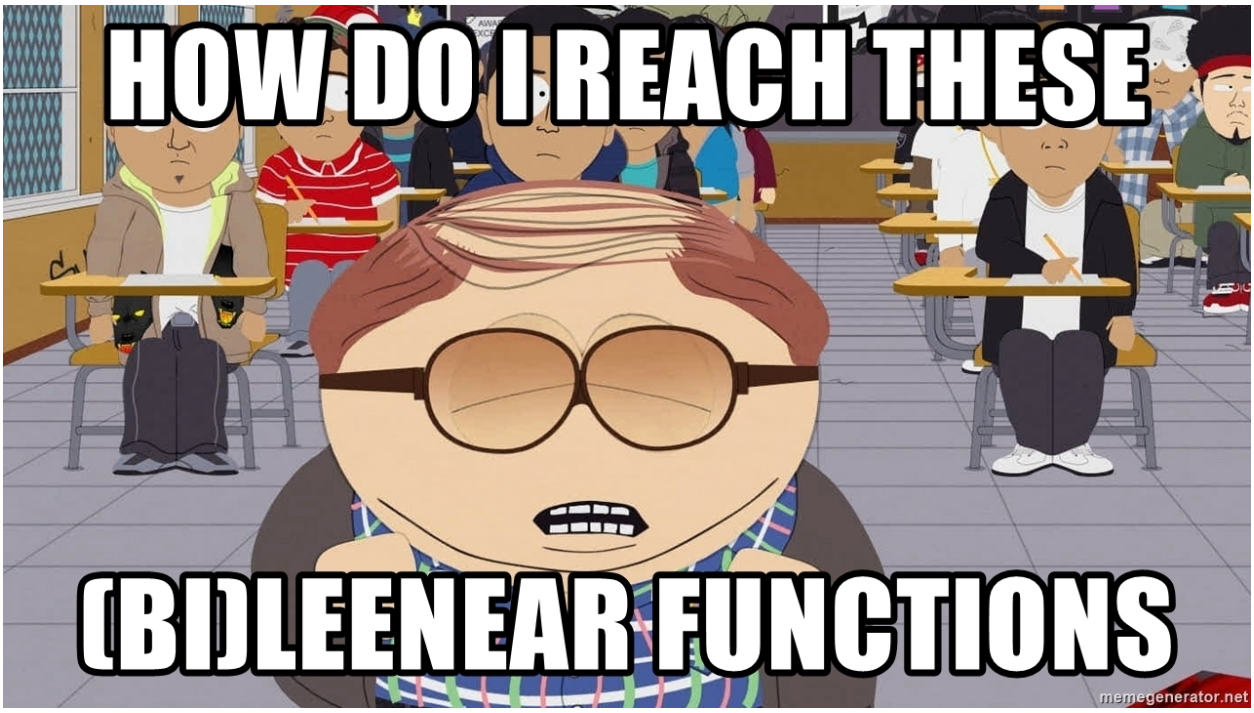
and try to compare it to the bilinear map

$$\begin{array}{rcl} \mathbb{F}_2^n \times \mathbb{F}_2^n & \longrightarrow & \mathbb{F}_2 \\ (x, y) & \longmapsto & \langle u \mid x \rangle \cdot \langle u \mid y \rangle \end{array}$$

Indeed, linear maps $\mathbb{F}_2^p \longrightarrow \mathbb{F}_2$ and bilinear maps $\mathbb{F}_2^p \times \mathbb{F}_2^p \longrightarrow \mathbb{F}_2$ have an attractive property in common: they satusfy **very strong separation properties**. Two linear maps (or two bilinear maps) are **either perfectly equal or substantially different**:

- two linear maps $\alpha : \mathbb{F}_2^p \longrightarrow \mathbb{F}_2$ and $\beta : \mathbb{F}_2^p \longrightarrow \mathbb{F}_2$ are either equal, or disagree on half of the inputs, i.e. if we set $\mathscr{E}(\alpha, \beta) = \{x \in \mathbb{F}_2^p \mid \alpha(x) = \beta(x)\}$, the set where α and β achieve equality, then:
  - if $\alpha = \beta$, then the subset $\mathscr{E}(\alpha, \beta)$ is full : $\dfrac{\#\mathscr{E}(\alpha, \beta)}{\#\mathbb{F}_2^p} = 1$,
  - if $\alpha \neq \beta$, then $\dfrac{\#\mathscr{E}(\alpha, \beta)}{\#\mathbb{F}_2^p} = \dfrac{1}{2}$;
- two bilinear maps $\varphi : \mathbb{F}_2^q \times \mathbb{F}_2^q \longrightarrow \mathbb{F}_2$ and $\psi : \mathbb{F}_2^q \times \mathbb{F}_2^q \longrightarrow \mathbb{F}_2$ are either equal, or disagree on at least $\frac{1}{4}$-ths of the inputs, i.e. if we set $\mathscr{E}(\varphi, \psi) = \{(x, y) \in \mathbb{F}_2^p \times \mathbb{F}_2^p \mid \varphi(x, y) = \psi(x, y)\}$, the set of pairs where φ and ψ achieve equality, then:
  - if $\varphi = \psi$, then $\mathscr{E}(\varphi, \psi)$ is is full : $\dfrac{\#\mathscr{E}(\varphi, \psi)}{\#(\mathbb{F}_2^q \times \mathbb{F}_2^q)} = 1$,
  - if $\varphi \neq \psi$, then $\dfrac{\#\mathscr{E}(\varphi, \psi)}{\#(\mathbb{F}_2^q \times \mathbb{F}_2^q)} \leq \dfrac{3}{4}$.

Thus, if we could run some probabilistic equality tests on these objects, there would be no close calls : tests would either work flawlessly, or fail quite spectacularly. So ony might wonder: is there a way to have these well behaved objects talk to us, and do work for us, given that we can only query f and g?



It turns out that there are ways of achieving this, as will be explained in the next two sections.

Test #2 : probabilistically testing if two implicitely defined vectors u and $\mathbf{U}$ satisfy $\mathbf{U} = u \otimes u$

> Suppose f and g are ρ-quasi-linear maps (given by their values Γ_f and Γ_g) with f is δ-close to a unique linear map $\langle u \mid - \rangle$, and g is δ-close to a unique linear map $\langle \mathbf{U} \mid - \rangle$. We describe a probabilistic test for $\mathbf{U} \stackrel{?}{=} u \otimes u$.

Let us assume that :

1. f is δ-close to some linear map $\langle u \mid - \rangle$ for some vector $u \in \mathbb{F}_2^n$
2. g is δ-close to some linear map $\langle \mathbf{U} \mid - \rangle$ for some vector $u \in \mathbb{F}_2^n \otimes \mathbb{F}_2^n \simeq \mathbb{F}_2^{n^2}$

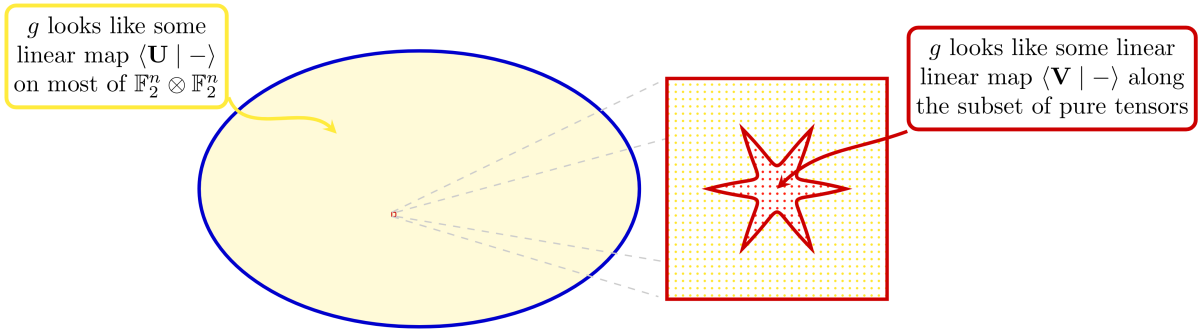
Naïve Test 2 (that fails)

Remembering the separation property for bilinear maps, one should be tempted to try a variation on the the following test:

1. Pick two points x and y in $\mathbb{F}_2^n$ uniformly at random.
2. Query Γ_g at $x \otimes y$.
3. Query Γ_f at x and y.
4. Compare $g(x \otimes y)$ and $f(x)f(y)$; `Reject` if they disagree.
5. Repeat $\mathsf{R} = C \cdot \lceil \frac{1}{\delta} \rceil$ many times and `Accept`.

(for some positive constant C to be determined). But this test doesn't work. It only queries g along the minuscule subset S of all pure tensors: $S = \{x \otimes y \mid x, y \in \mathbb{F}_2^n\} \subset \mathbb{F}_2^n \otimes \mathbb{F}_2^n$, a set of size $\#S = 1 + (2^n - 1)^2 = 2^{2n} - 2^{n+1} + 2 \ll 2^{n^2} = \#(\mathbb{F}_2^n \otimes \mathbb{F}_2^n)$. A dishonest prover could try to exploit this by defining g to behave like the linear map $\langle \mathbf{U} \mid - \rangle$ along $\mathbb{F}_2^n \otimes \mathbb{F}_2^n \setminus S$, and like $\langle \mathbf{V} \mid - \rangle$ along S. For large enough S this defines

a ρ-quasi-linear map. For instance it could choose $\mathbf{U}$ to be a solution of $A\mathbf{U} = b$, and $\mathbf{V} = u \otimes u$ for an *arbitrary* $u \in \mathbb{F}_2^n$.



Fixing the naïve test

> To fix the previous test we must change it so that we explore more of g's domain than the tiny subset S of all pure tensors $x \otimes y$. To achieve this, we will query g twice rather than once, and exploit the fact that it behaves like a linear map on most inputs, so that, **more often than not**, $g(x \otimes y) = g(x \otimes y + r) + g(r)$ where x, y and r are drawn uniformly at random from $\mathbb{F}_2^n$, $\mathbb{F}_2^n$ and $\mathbb{F}_2^n \otimes \mathbb{F}_2^n$ respectively. This way we can explore the full domain of g by queriying it at $x \otimes y + r$ and r.

For reasons that will become clear as we go, let's take $\delta < \frac{1}{20}$, hence $\frac{3}{4} + 5\delta < 1$. Let us define the following :

$$L_f = \left\{ x \in \mathbf{F}_2^n \;\middle|\; f(x) = \langle u \mid x \rangle \right\}$$

$$L_g = \left\{ x \in \mathbf{F}_2^n \otimes \mathbf{F}_2^n \;\middle|\; g(x) = \langle \mathbf{U} \mid x \rangle \right\}$$

By virtue of the assumption f and g being ρ-quasi-linear, by definition of u and $\mathbf{U}$ and the theorem — **REFERENCE PLEASE**—,

$$\frac{\#L_f}{\#\mathbb{F}_2^n} \geq 1 - \delta \quad \text{and} \quad \frac{\#L_g}{\#(\mathbb{F}_2^n \otimes \mathbb{F}_2^n)} \geq 1 - \delta. \tag{1}$$

We know from the separation properties of bilinear maps — **REFERENCE PLEASE**— that

$$u \otimes u = \mathbf{U} \iff \frac{\#\Big\{(x,y) \in \mathbf{F}_2^n \times \mathbf{F}_2^n \mid \langle \mathbf{U} \mid x \otimes y \rangle = \langle u \mid x \rangle \langle u \mid y \rangle\Big\}}{\#\Big(\mathbf{F}_2^n \times \mathbf{F}_2^n\Big)} > \frac{3}{4}$$

We will want to test for this condition. We of course don't have direct access to u and $\mathbf{U}$, so we need a workaround. Consider three independent uniformly distributed random variables X, Y with values in $\mathbb{F}_2^n$, and R with values in $\mathbb{F}_2^{n^2}$. Then $X \otimes Y + R$ is also uniformly distributed over $\mathbb{F}_2^{n^2}$. Also, let us set

$$T = \left\{(x,y,r) \in \mathbf{F}_2^n \times \mathbf{F}_2^n \times (\mathbf{F}_2^n \otimes \mathbf{F}_2^n) \;\middle|\; g(x \otimes y + r) + g(r) = f(x) f(y)\right\}$$

We see that

$$\begin{aligned}
\big[\langle \mathbf{U} \mid X \otimes Y \rangle = \langle u \mid X \rangle \langle u \mid Y \rangle\big] \quad &\supset \quad \left\{ \begin{array}{l} \big[\langle u \mid X \rangle = f(X)\big] \\ \cap \, \big[\langle u \mid Y \rangle = f(Y)\big] \\ \cap \, \big[\langle \mathbf{U} \mid X \otimes Y + R \rangle = g(X \otimes Y + R)\big] \\ \cap \, \big[\langle \mathbf{U} \mid R \rangle = g(R)\big] \end{array} \right\} \cap \, \big[g(X \otimes Y + R) + g(R) = f(X) f(Y)\big] \\
&\supset \quad \underbrace{\left\{ \begin{array}{l} \big[X \in L_f\big] \\ \cap \, \big[Y \in L_f\big] \\ \cap \, \big[X \otimes Y + R \in L_g\big] \\ \cap \, \big[R \in L_g\big] \end{array} \right\}}_{=B} \cap \, \big[(X,Y,R) \in T\big] \\
&= \quad \big[(X,Y,R) \in T\big] \, \cap \, B \\
&= \quad \big[(X,Y,R) \in T\big] \, \setminus \, B^c
\end{aligned}$$

Now note that the complement of B breaks down as a union :

$$\begin{aligned}
B^c \quad = \quad & \big[X \notin L_f\big] \\
& \cup \big[Y \notin L_f\big] \\
& \cup \big[X \otimes Y + R \notin L_g\big] \\
& \cup \big[R \notin L_g\big]
\end{aligned}$$

so that when we take probabilities

$$\begin{aligned}\mathbb{P}(B^c) &\leq \mathbb{P}\Big(\big[X \notin L_f\big]\Big) \\ &\quad + \mathbb{P}\Big(\big[Y \notin L_f\big]\Big) \\ &\quad + \mathbb{P}\Big(\big[X \otimes Y + R \notin L_g\big]\Big) \\ &\quad + \mathbb{P}\Big(\big[R \notin L_g\big]\Big) \\ &\leq 4\delta\end{aligned}$$

and thus

$$\begin{aligned}\mathbb{P}\Big(\big[\langle \mathbf{U} \mid X \otimes Y\rangle = \langle u \mid X\rangle\langle u \mid Y\rangle\big]\Big) &\geq \mathbb{P}\Big(\big[(X,Y,R) \in T\big]\Big) - \mathbb{P}\Big(B^c\Big) \\ &\geq \mathbb{P}\Big(\big[(X,Y,R) \in T\big]\Big) - 4\delta\end{aligned}$$

The probability on the left hand side is precisely the quantity we are interested in:

$$\mathbb{P}\Big(\big[\langle \mathbf{U} \mid X \otimes Y\rangle = \langle u \mid X\rangle\langle u \mid Y\rangle\big]\Big) = \frac{\#\Big\{(x,y) \in \mathbf{F}_2^n \times \mathbf{F}_2^n \mid \langle \mathbf{U} \mid x \otimes y\rangle = \langle u \mid x\rangle\langle u \mid y\rangle\Big\}}{\#\Big(\mathbf{F}_2^n \times \mathbf{F}_2^n\Big)}$$

and so our computation amounts to

$$\frac{\#\Big\{(x,y) \in \mathbf{F}_2^n \times \mathbf{F}_2^n \mid \langle \mathbf{U} \mid x \otimes y\rangle = \langle u \mid x\rangle\langle u \mid y\rangle\Big\}}{\#\Big(\mathbf{F}_2^n \times \mathbf{F}_2^n\Big)} \geq \frac{\#T}{\#(\mathbf{F}_2^n \times \mathbf{F}_2^n \times (\mathbf{F}_2^n \otimes \mathbf{F}_2^n))} - 4\delta$$

Since our goal is to probe whether the purple term on the left is indeed $> \frac{3}{4}$, it is enough for us to probe whether $\frac{\#T}{\#(\mathbb{F}_2^n \times \mathbb{F}_2^n) \times (\mathbb{F}_2^n \otimes \mathbb{F}_2^n))} - 4\delta \stackrel{?}{>} \frac{3}{4}$. Now is where the condition that δ satisfy $\frac{3}{4} + 5\delta < 1$ becomes important:

$$\left[\frac{\#T}{\#(\mathbf{F}_2^n \times \mathbf{F}_2^n) \times (\mathbf{F}_2^n \otimes \mathbf{F}_2^n)} > \mathbf{1 - \delta}\right] \Longrightarrow \left[\frac{\#\Big\{(x,y) \in \mathbf{F}_2^n \times \mathbf{F}_2^n \mid \langle \mathbf{U} \mid x \otimes y\rangle = \langle u \mid x\rangle\langle u \mid y\rangle\Big\}}{\#\Big(\mathbf{F}_2^n \times \mathbf{F}_2^n\Big)} > \frac{3}{4}\right]$$

(indeed, $\frac{3}{4} + 4\delta < 1 - \delta$) and we can finally test for the following condition:

$$\frac{\#T}{\#(\mathbf{F}_2^n \times \mathbf{F}_2^n) \times (\mathbf{F}_2^n \otimes \mathbf{F}_2^n)} \overset{?}{>} 1 - \delta$$

We apply the following test:

Test for probabilistically ruling out small subsets

If $S \subset X$ is a subset of a finite set X, if $0 < \delta < 1$, then we can probabilistically check whether S is small, in the sense that $\frac{\#S}{\#X} \leq \rho = 1 - \delta$:

1. Pick $x \in X$ uniformly at random;
2. test whether $x \in S$; `Reject` if not;
3. repeat $\mathsf{R} = \lceil \frac{\delta}{\delta} \rceil$ times and `Accept`

For $\frac{\#S}{\#X} \leq 1 - \delta$, his test succeeds with probability $\leq \frac{1}{e^\delta}$, and succeeds 100% of the time if $S = X$. In particular, if we take $x = 1$, then this test rejects subsets $S \subset X$ such that $\frac{\#S}{\#X} \leq 1 - \delta$ with probability $> 50\%$.

Improved Test 2

1. Pick two points x and y in $\mathbb{F}_2^n$ uniformly at random, and pick $r \in \mathbb{F}_2^n \otimes \mathbb{F}_2^n$ uniformly at random.
2. Query Γ_g at $x \otimes y + r$ and r.
3. Query Γ_f at x and y.
4. Compare $g(x \otimes y + r) + g(r)$ and $f(x)f(y)$; `Reject` if they disagree.
5. Repeat $\mathsf{R} = \lceil \frac{1}{\delta} \rceil$ many times and `Accept`.

If f and g pass this test, then we can state with probability $> 50\%$, that $\mathbf{U} = u \otimes u$.

Test #3 : probabilistically testing if an implicitely defined vector U satisfies the linear system $A\mathbf{U} = b$

> This is the simplest of the three tests. But even so we run into a small problem, not too dissimilar from the preceding one : there is a naïve test, but doesn't explore enough of de graph of g. Thankfully, the fix is easier this time around.

We assume that we are given a $1 - \delta = \rho$-quasi-linear map $g : \mathbb{F}_2 \otimes \mathbb{F}_2 \to \mathbb{F}_2$ given by its values Γ_g. Let $\mathbf{U} \in \mathbb{F}_2^n \otimes \mathbb{F}_2^n$ represent the unique closest linear map to g. Recall from — **REFERENCE PLEASE** —, if two vectors a and b in $\mathbb{F}_2^p$ are distinct, then for half of all $x \in \mathbb{F}_2^p$, $\langle a \mid x \rangle \neq \langle b \mid x \rangle$. Thus, the two vectors of interest $A\mathbf{U}$ and b are the same if an donly if

$$A\mathbf{U} = b \iff \frac{\#\left\{x \in \mathbb{F}_2^m \mid \langle A\mathbf{U} \mid x\rangle = \langle b \mid x\rangle\right\}}{\#\mathbb{F}_2^m} > \frac{1}{2}$$

Note that $\langle A\mathbf{U} \mid x\rangle = \langle \mathbf{U} \mid {}^tAx\rangle$. A naïve test for $A\mathbf{U} \stackrel{?}{=} b$ could run like so :

Naïve Test 3 (that fails)

1. Pick a point $x \in \mathbb{F}_2^m$ uniformly at random and compute ${}^tAx$.
2. Query Γ_g at ${}^tAx$.
3. Compare $g({}^tAx)$ and $\langle b \mid x\rangle$; `Reject` if they disagree.
4. Repeat R many times and `Accept`.

Again, there is a problem with this approach: the "test vectors" ${}^tAx$ only run through the subspace $\mathrm{Im}({}^tA) \subset \mathbb{F}_2^n \otimes \mathbb{F}_2^n$, and this opens the door to cheating the verifier. To be precise, the verifier could fix g so that along $\mathrm{Im}({}^tA)$ we have $g(y) = \langle b \mid x\rangle$ where $y = {}^tAx$, while outside of $\mathrm{Im}({}^tA)$, $g(y) = \langle \mathbf{U} \mid y\rangle$, for some *random* $\mathbf{U}$. If ${}^tA$ is one to one, as it will often be if derived from a problem in **3-SAT**, this Γ_g would be ρ-quasi-linear, pass this test, all the while $A\mathbf{U} \neq b$.

Fixing the naïve test

> We will use the same approach as previously : ask the Verifier to read from everywhere in the proof by adding a random parameter $r \in \mathbb{F}_2^n \otimes \mathbb{F}_2^n$ to the mix.

What follows is very much the same as what we did for the second test. We use the same notation L_g as previously. Again, $\frac{\#L_g}{\#(\mathbb{F}_2^n \otimes \mathbb{F}_2^n)} \geq \rho = 1 - \delta$. Again, we impose an upper bound on δ, this time we ask that it satisfy: $1 - 3\delta > \frac{1}{2}$, i.e. $\delta < \frac{1}{6}$. Let us set

$$T' = \left\{(x, r) \in \mathbb{F}_2^m \times (\mathbb{F}_2^n \otimes \mathbb{F}_2^n) \;\middle|\; g({}^tAx + r) + g(r) = \langle b \mid x\rangle\right\}$$

Let X be uniformly distributed random variable in $\mathbb{F}_2^n$, and let R be uniformly distributed random variable in $\mathbb{F}_2^n \otimes \mathbb{F}_2^n$, independent of X. We see that

$$\begin{aligned}
\left[\langle \mathbf{U} \mid {}^tAX\rangle = \langle b \mid X\rangle\right] &\supset \left\{\begin{array}{l}\left[\langle \mathbf{U} \mid {}^tAX + R\rangle = g({}^tAX + R)\right] \\ \cap\ \left[\langle \mathbf{U} \mid R\rangle = g(R)\right]\end{array}\right\} \cap \left[g({}^tAX + R) + g(R) = \langle b \mid X\rangle\right] \\
&\supset \underbrace{\left\{\begin{array}{l}\left[{}^tAX + R \in L_g\right] \\ \cap\ \left[R \in L_g\right]\end{array}\right\}}_{=B'} \cap \left[(X, R) \in T'\right] \\
&= \left[(X, R) \in T'\right] \cap B' \\
&= \left[(X, R) \in T'\right] \setminus B'^c
\end{aligned}$$

Now note that the complement of B breaks down as a union :

$$\begin{aligned} B'^c &= [{}^tAX + R \in L_g] \\ &\quad \cup \left[R \in L_g\right] \end{aligned}$$

so that when we take probabilities

$$\begin{aligned} \mathbb{P}(B'^c) &\leq \mathbb{P}([{}^tAX + R \in L_g]) \\ &\quad + \mathbb{P}(\left[R \in L_g\right]) \\ &\leq 2\delta \end{aligned}$$

and thus

$$\begin{aligned} \mathbb{P}\Big(\left[\langle \mathbf{U} \mid {}^tAX\rangle = \langle b \mid X\rangle\right]\Big) &\geq \mathbb{P}\Big(\left[(X,R) \in T'\right]\Big) - \mathbb{P}\Big(B'^c\Big) \\ &\geq \mathbb{P}\Big(\left[(X,R) \in T'\right]\Big) - 2\delta \end{aligned}$$

The probability on the left hand side is precisly the quantity we are interested in:

$$\mathbb{P}\Big(\left[\langle \mathbf{U} \mid {}^tAX\rangle = \langle b \mid X\rangle\right]\Big) = \frac{\#\Big\{x \in \mathbb{F}_2^m \mid \langle A\mathbf{U}|x\rangle = \langle b|x\rangle\Big\}}{\#\mathbb{F}_2^m}$$

and so our computation amounts to

$$\frac{\#\left\{x\in\mathbb{F}_2^m \,|\, \langle AU|x\rangle = \langle b|x\rangle\right\}}{\#\mathbb{F}_2^m} \geq \frac{\#T'}{\#\left(\mathbb{F}_2^m \times (\mathbb{F}_2^n \otimes \mathbb{F}_2^n)\right)} - 2\delta$$

Since our goal is to probe whether the purple term on the left is indeed $> \frac{1}{2}$, it is enough for us to probe whether $\frac{\#T}{\#(\mathbb{F}_2^n \times \mathbb{F}_2^n) \times (\mathbb{F}_2^n \otimes \mathbb{F}_2^n))} - 2\delta \overset{?}{>} \frac{1}{2}$. Now is where the condition that δ satisfy $\frac{1}{2} < 1 - 3\delta$ becomes important, because it implies that

$$\left[\frac{\#T'}{\#\left(\mathbb{F}_2^m \times (\mathbb{F}_2^n \otimes \mathbb{F}_2^n)\right)} > 1 - \delta\right] \Longrightarrow \left[\frac{\#\left\{x\in\mathbb{F}_2^m \,|\, \langle AU|x\rangle = \langle b|x\rangle\right\}}{\#\mathbb{F}_2^m} > \frac{1}{2}\right]$$

(indeed, $(1-\delta) - 2\delta < \frac{1}{2}$). We can use this to test for the following condition:

$$\frac{\#T'}{\#\left(\mathbb{F}_2^m \times (\mathbb{F}_2^n \otimes \mathbb{F}_2^n)\right)} \overset{?}{>} 1 - \delta$$

The following is a test for this condition. We apply the test for probabilistically ruling out small subsets described in — **REFERENCE PLEASE** —

Improved Test 2

1. Pick two points $x, r \in \mathbb{F}_2^m$ uniformly at random.
2. Compute the dot product $\langle b \mid x\rangle$ (polynomial time in m).
3. Compute ${}^tAx$ (polynomial time in m and n) and ${}^tAx + r$ (polynomial time in n).
4. Query from Γ_g the values $g({}^tAx + r)$ and $g(r)$.
5. Compare $g({}^tAx + r) + g(r)$ and $\langle b \mid x\rangle$.
6. `Reject` if they disagree.
7. Repeat R-many times and `Accept`.

Tying everything together: the verifier Test

We describe, at last, the test the verifier applies toPCP proofs. This combines the three tests previously described.

Description of the verifier test

Phase 0 - Setup

The verifier is provided with an instance (A, b) of **QUADEQ** (or an instance φ of **3-SAT** which the verifier and the prover convert to an equivalent instance of **QUADEQ**). The verifier is given access to a PCP proof (Γ_f, Γ_g) and it wants to verify it. The verifier fixes some numerical value for $\delta < \frac{1}{20}$, say $\delta = 0.01$ and $\rho = 1 - \delta = 0.99$.

Phase 1 - Quasi-linearity testing

The **first** test the verifier applies to the PCP proof is the linearity test from —**REFERENCE PLEASE**—:

- run $\mathcal{T}_\delta$ 4 times on Γ_f
- run $\mathcal{T}_\delta$ 4 times on Γ_g

If both f and g pass these tests, the verifier moves to the next round of testing.

Phase 2 - "$\mathbf{U} = u \otimes u$" testing

The **second** test the verifier applies to the PCP proof is the linearity test from —**REFERENCE PLEASE**—: it runs this test, say, 4 times. If the proof (Γ_f, Γ_g) passes this test, it moves to the third and final round.

Phase 3 - "$A\mathbf{U} = b$" testing

The **third** test the verifier applies to the PCP proof is the linearity test from —**REFERENCE PLEASE**—: it runs this test, say, 4 times. If the proof passes this final test, the verifier accepts.

Complexity Analysis

Phase 1

Query Complexity: phase 1 requires $4 \times \frac{1}{\delta} \times 6 = 2400$ queries to the proof. Indeed, we do 4 repetitions, every run takes $\frac{1}{\delta} = 100$ rounds, and at every round we have to make 3 queries to f ($f(x)$, $f(y)$ and $f(x+y)$) and 3 queries to g ($g(x')$, $g(y')$ and $g(x'+y')$).

Randomness Complexity: $4 \times \frac{1}{\delta} = 400$ times we have to

- create random $x, y \in \mathbb{F}_2^n$, which requires $2n$ bits of randomness
- create random $x', y' \in \mathbb{F}_2^n \otimes \mathbb{F}_2^n$, which requires $2n^2$ bits of randomness.

This makes for $400 \times (2n + 2n^2)$ bits of randomness.

Computational Complexity: $4 \times \frac{1}{\delta} = 400$ times we have to

- add the vectors x and y (both have length n)
- add the bits $f(x)$ to $f(y)$

- compare the result to $f(x+y)$

This makes for $400 \times (n+1+1)$ bit operations. Similarly, 400 times have to

- add the vectors x' and y' (both have length n^2)
- add the bits $g(x)$ to $g(y)$
- compare the result to $g(x+y)$.

This makes for $400 \times (n^2+1+1)$ further bit operations.

Phase 2

Query Complexity: phase 2 requires $4 \times \frac{1}{\delta} \times 4 = 1600$ queries to the proof: we do 4 repetitions, every run takes $\frac{1}{\delta} = 100$ rounds, and at every round we have to make 4 queries: $g(x \otimes y + r)$, $g(r)$, $f(x)$ and $f(y)$

Randomness Complexity: $4 \times \frac{1}{\delta} = 400$ times we have to

- create random $x, y \in \mathbb{F}_2^n$, which requires $2n$ bits of randomness;
- create random a random $r \in \mathbb{F}_2^n \otimes \mathbb{F}_2^n$, which requires n^2 bits of randomness.

This makes for $400 \times (2n + n^2)$ bits of randomness

Computational Complexity: $4 \times \frac{1}{\delta} = 400$ times we have to

- compute the n^2 coordinates of $x \otimes y$ (each bit of $x \otimes y$ is the product of two bits of x and y)
- compute the sum $x \otimes y + r$, which requires n^2 bit operations
- multiply the bits $f(x)$ and $f(y)$
- add the bits $g(x \otimes y + r)$ and $g(r)$
- compare the bits $g(x \otimes y + r) + g(r)$ and $f(x)f(y)$.

This makes for $400 \times (n^2 + n^2 + 1 + 1 + 1)$ bit operations.

Phase 3

Query Complexity: phase 3 requires $4 \times \frac{1}{\delta} \times 2 = 800$ queries to the proof: we do 4 repetitions, every run takes $\frac{1}{\delta} = 100$ rounds, and at every round we have to make 2 queries: $g({}^tAx + r)$ and $g(r)$.

Randomness Complexity: $4 \times \frac{1}{\delta} = 400$ times we have to

- create random $x \in \mathbb{F}_2^m$, which requires $2m$ bits of randomness;
- create random a random $r \in \mathbb{F}_2^n \otimes \mathbb{F}_2^n$, which requires n^2 bits of randomness.

This makes for $400 \times (m + n^2)$ bits of randomness

Computational Complexity: $4 \times \frac{1}{\delta} = 400$ times we have to

- compute the n^2 coordinates of ${}^tAx$, which requires $m \times 2 \times n^2$ bit operations
- compute the sum ${}^tAx + r$, which requires n^2 bit operations
- compute the dot product $\langle b \mid x \rangle$, which requires $2m$ operations
- add the bits $g({}^tAx + r)$ and $g(r)$
- compare the bits $g({}^tAx + r) + g(r)$ and $\langle b \mid x \rangle$.

This makes for $400 \times (2 \times n^2 \times m + n^2 + 2m + 1 + 1)$ bit operations.

Conclusion

Thus the **total query complexity** is constant: we only need to query the proof 4800 times, **the number of random bits we need** is polynomial in the complexity (say) $n + m$ of the instance (A, b), and the **total computational complexity** is also polynomial in the complexity of the instance.

Soundness Analysis

If (A, b) is a YES instance of QUADEQ

Then if the prover knows a witness u, it can compute the graphs $\Gamma_f = \Gamma_{\langle u|-\rangle}$ and $\Gamma_g = \Gamma_{\langle u\otimes u|-\rangle}$. The proof (Γ_f, Γ_g) passes the verifier's test without fault 100% of the time.

If (A, b) is a NO instance of QUADEQ

We need to show that **whatever** "proof" (Γ_f, Γ_g) is passed to the verifier, it rejects it at least 50% of the time. So let $\pi = (\Gamma_f, \Gamma_g)$ be such a "proof". We consider different possibilities for π.

- If either of Γ_f or Γ_g is *not* ρ-quasi-linear, then π passes the first round of testing with probability $\leq \frac{1}{2^4} \cdot \frac{1}{2^4}$, i.e. fails with probability at least $\frac{255}{256}$.
- If both Γ_f and Γ_g are ρ-quasi-linear, and the Hamming distance closest linear maps are represented by vectors u and $\mathbf{U}$ respectively, but the relationship $\mathbf{U} = u \otimes u$ is **false**, then π passes the second round of testing with probability $\leq \frac{1}{2^4}$, i.e. fails with probability $\geq \frac{15}{16}$.
- If both Γ_f and Γ_g are ρ-quasi-linear, and closest linear maps are represented by vectors u and $\mathbf{U}$ satisfying $\mathbf{U} = u \otimes u$, then by virtue of the fact that (A, b) is a NO instance of **QUADEQ**, the equation $A\mathbf{U} = b$ **cannot hold**. Thus π passes the third round of testing with probability $\leq \frac{1}{2^4}$, i.e. fails with probability $\geq \frac{15}{16}$.

Now the supposed proof π falls into one of those three categories. Therefore, the probability for that proof of passing the verifier is

$$\mathbb{P}\big(\pi \text{ is accepted}\big) \leq \frac{1}{2^4} \cdot \frac{1}{2^4} + \frac{1}{2^4} + \frac{1}{2^4} < \frac{1}{4}$$

We have thus established that this PCP verifier for **QUADEQ** (and by extension **3-SAT**) satisfies the required soundness:

1. **Completeness:** if (A, b) is a YES instance of **QUADEQ**, there exists a proof π such that $\mathbb{P}[V^\pi((A, b)) = 1] = 1$;
2. **Soundness:** if (A, b) is a NO instance of **QUADEQ**, then for every proof π, $\mathbb{P}[V^\pi((A, b)) = 1] \leq \frac{1}{2}$.