The Tensor Product of Two Codes is Not Necessarily Robustly Testable

Paul Valiant

Massachusetts Institute of Technology
pvaliant@mit.edu

Abstract. There has been significant interest lately in the task of constructing codes that are testable with a small number of random probes. Ben-Sasson and Sudan show that the repeated tensor product of codes leads to a general class of locally testable codes. One question that is not settled by their work is the local testability of a code generated by a single application of the tensor product. Special cases of this question have been studied in the literature in the form of "tests for bivariate polynomials", where the tensor product has been shown to be locally testable for certain families of codes. However the question remained open for the tensor product of generic families of codes. Here we resolve the question negatively, giving families of codes whose tensor product does not have good local testability properties.

1 Introduction

Sub-linear algorithms in coding theory have become an increasingly important area of study, both for their own sake, and because of their strong connection with the techniques of the PCP program. A key problem is that of constructing and understanding Locally Testable Codes (LTCs), namely codes for which membership can be estimated probabilistically using a sub-linear number of queries (cf. [1, 11, 7, 8]).

The tensor product code of two linear codes C_1, C_2 of length n, denoted $C_1 \otimes C_2$, is the code consisting of the set of all $n \times n$ matrices in which each row belongs to C_1 and each column to C_2. The prime example of a tensor code is the set of bivariate polynomials of individual degree k over a field of size n, which is the tensor product of a pair of (univariate) Reed-Solomon codes of degree k.

A basic result of Arora and Safra says that if an $n \times n$ matrix is δ-far from the above-mentioned bivariate code then the expected distance of a random row/column of the matrix from a univariate degree k polynomial for $k = O(n^{\frac{1}{3}})$ is $\Omega(\delta)$ [3]. (A similar result with linear relationship between k and n was shown by Polishchuk and Spielman [10].) Formally, this means that the transformation from the univariate codes to the bivariate product code is *robust*, as distance from the product code implies similar expected distance from the component Reed-Solomon codes.

Recently, Ben-Sasson and Sudan, using ideas from Raz and Safra [12], showed that the slightly more complicated three-wise tensor product code is robustly locally testable under much more general conditions than the above polynomial restriction [5].

Robust locally testable codes are crucial in efficient constructions of LTCs and in fact many PCPs (see [5, 6]). In light of the above two results, it is natural to ask whether the tensor product of pairs of general codes is robustly locally testable. The current paper gives a negative answer to this basic question.

2 Definitions

We begin with some basic notions from coding theory. The first is the *Hamming* metric on strings.

Definition 1 (Hamming distance). *Given a set Σ called the* alphabet*, and two strings $x, y \in \Sigma^n$ for some n, denote the ith entry of each string by x_i, y_i respectively. The Hamming distance between x and y, denoted $\Delta(x, y)$, is the number of indices i for which $x_i \neq y_i$.*

A *code* is a means of transforming strings into (typically) longer strings, such that the original message is recoverable from the *encoded* message even when the encoded message has been tampered with by some bounded adversary. This notion is made precise with the following definition.

Definition 2 (Code). *Given an alphabet Σ, and integers (n, k, d), referred to respectively as the* block length*, the* message length*, and the* distance*, an $[n, k, d]_\Sigma$ code is defined by an encoding function*

$$f : \Sigma^k \to \Sigma^n$$

such that Hamming distance between any two elements of the image of f is at least d.

A code C is often identified with the image of its encoding function

$$C \stackrel{\text{def}}{=} \{f(x) | x \in \Sigma^k\}.$$

The elements of C are referred to as codewords*.*

We define a few more notions related to the Hamming distance that are frequently used in coding theory.

Definition 3 (Distance notions). *The ratio of the Hamming distance to the length of strings n is called the* relative distance*, and is denoted by $\delta(x, y) \stackrel{\text{def}}{=} \Delta(x, y)/n$. Given a code—or alternatively any set of strings—C, we define $\Delta_C(x), \delta_C(x)$ to be respectively, the minimum Hamming distance and relative distance between x and any codeword of C. Similarly, let $\Delta(C)$ be the minimum Hamming distance between any two distinct codewords of C, and let $\delta(C) \stackrel{\text{def}}{=} \Delta(C)/n$.*

In this paper we consider *linear codes*, defined as follows:

Definition 4 (Linear Code). *An $[n, k, d]_\Sigma$ linear code is an $[n, k, d]_\Sigma$ code where Σ is a field, and the encoding function f is linear. We represent f by a $k \times n$ matrix M, known as the* generator *matrix, with the property that $f(x) = x^T M$.*

Alternatively, an $[n, k, d]_\Sigma$ linear code C is a k-dimensional linear subspace of Σ^n such that $\Delta(C) \geq d$.

Given a code C, and a string $x \in \Sigma^n$, we often wish to determine if x is a codeword of C and, if not, what the closest codeword of C to x is. Recently, the focus has changed from deterministic tests that look at x in its entirety, to probabilistic tests that sample x at only a few locations. Such tests are called *local tests* (cf. [5]).

Definition 5 (Local Tester). *A tester T with query complexity q is a probabilistic oracle machine that when given oracle access to a string $r \in \Sigma^n$, makes q queries to the oracle for r and returns an accept/reject verdict. We say that T tests a code C of minimum distance d if whenever $r \in C$, T accepts with probability one; and when $r \notin C$, the tester rejects with probability at least $\delta_C(r)/2$. A code C is said to be locally testable with q queries if C has minimum distance $d > 0$ and there is a tester for C with query complexity q.*

A generic way to create codes that have non-trivial local tests is via the tensor product. The next two definitions describe the tensor product and a natural local test for codes constructed by this product (cf. [9], [13, Lecture 6, Sect. 2.4]).

Definition 6 (Tensor Product Code). *Given an $[n_1, k_1, d_1]_\Sigma$ code C_1, and an $[n_2, k_2, d_2]_\Sigma$ code C_2, define their* tensor product*, denoted $C_1 \otimes C_2 \subseteq \Sigma^{n_2 \times n_1}$, to be the set of $n_2 \times n_1$ matrices each of whose rows is an element of C_1, and each of whose columns is an element of C_2. It is well known that $C_1 \otimes C_2$ is an $[n_1 n_2, k_1 k_2, d_1 d_2]_\Sigma$ code.*

We note that if C_1, C_2 are linear *codes, with corresponding generator matrices G_1, G_2, then the set of matrices with rows in the row-space of G_1 and columns in the row-space of G_2 is just the set*

$$C_1 \otimes C_2 \stackrel{\text{def}}{=} \{M_2^T X M_1 | X \in \Sigma^{k_2 \times k_1}\}.$$

The natural probabilistic test for membership in a product code C is the following.

Definition 7 (Product Tester). *Given a product code $C = C_1 \otimes C_2$, test a matrix r for membership in C as follows: flip a coin; if it is heads, test whether a random row of r is a codeword of C_1; if it is tails, test whether a random column of r is a codeword of C_2.*

It is straightforward to show that in fact this product tester meets the criteria of Definition 5 for a local tester (see the appendix).

Thus we see that the tensor product of two codes of length n has a local test with query complexity $\sqrt{N}$, where $N = n^2$ is the length of the tensored code.

We now ask whether the local tests for the tensor product may be recursively applied. Specifically, if we take the tensor product of two tensor product codes $C^4 = C^2 \otimes C^2 = C \otimes C \otimes C \otimes C$, we know we can locally test for membership in C^4 by randomly testing for membership in C^2; could we now save time on this test by applying a local test to C^2 instead of a more expensive deterministic test?

It turns out that locally testable codes do not necessarily compose in this way without an additional *robustness* property. Motivated by the notion of Robust PCP verifiers introduced in [4], Ben-Sasson and Sudan introduce the notion of a *robust* local tester (cf. [5]). Informally a test is said to be robust if words that are far from codewords are not only rejected by the tester with high probability, but in fact, with high probability, the *view* of the tester is actually far from any accepting *view*. This is defined formally as follows.

Definition 8 (Robust Local Tester). *A tester T has two inputs: an oracle for a received vector r, and a random string s. On input the string s the tester generates queries $i_1, \ldots, i_q \in [n]$ and fixes circuit $C = C_s$ and accepts if $C(r[i_1], \ldots, r[i_q]) = 1$. For oracle r and random string s, define the robustness of the tester T on r, s, denoted $\rho^T(r, s)$, to be the minimum, over strings x satisfying $C(x) = 1$, of relative distance of $\langle r[i_1], \ldots, r[i_q] \rangle$ from x. We refer to the quantity $\rho^T(r) \stackrel{\text{def}}{=} \mathbf{E}_s[\rho^T(r, s)]$ as the expected robustness of T on r.*

A tester T is said to be α-robust for a code C if for every $r \in C$, the tester accepts w.p. one, and for every $r \in \Sigma^n$, $\rho^T(r) \geq \alpha \cdot \delta_C(r)$.

This definition is especially meaningful for the product tester of Definition 7. In this case, we may interpret the above definition of robustness as follows. Given a tensor product code $C = C_1 \otimes C_2$, the queries $i_1, ..., i_q$ comprise either a row or a column of the matrix r. For random seed s, denote this row or column by r^s. We note that if r^s is a row, then the robustness $\rho^T(r, s)$ is just the relative distance of r^s from the nearest codeword of C_1, namely $\delta_{C_1}(r^s)$, while if r^s is a column, then $\rho^T(r, s) = \delta_{C_2}(r^s)$. Thus the expected robustness of T on r is just the average of the following two quantities: the average relative distance of rows of r from C_1; and the average relative distance of columns of r from C_2. A robust tester signifies that any time r is close to C_1 row-wise, and close to C_2 column-wise, then it is also close to a single element of C with rows in C_1 and columns in C_2.

One of the few known examples of robust locally testable codes are codes based on multivariate polynomials. Codes based on multivariate polynomials may in turn be viewed as tensor products of (univariate) Reed-Solomon codes, which are defined as follows. Given an alphabet Σ that is a finite field $\mathbb{F}_q$, we encode degree k polynomials over this field by their values on $n > k$ fixed elements $\alpha_1, ..., \alpha_n$ of $\mathbb{F}_q$.

The tensor product of two Reed-Solomon codes is a *bivariate* code, which represents a bivariate polynomial by its values on a "rectangle" in $\mathbb{F}_q \times \mathbb{F}_q$.

Local testability—and in fact robustness—of the product tester for the bivariate polynomial codes is well-studied in the literature on PCPs. Results of Polishchuk and Spielman (cf.[10, Theorem 9]) imply that for bivariate codes $C = C_1 \otimes C_1$, there exist $\epsilon > 0, c < \infty$ such that the product tester of above is in fact an ϵ-robust local tester provided the parameters n, k of C_1 satisfy $n \geq ck$. Similar results are implied by Lemma 5.2.1 of [3], specialized to the case $m = 2$. The resulting robustness analysis of the bivariate test is critical to many PCPs in the literature including those of [3, 2, 10, 6].

The robustness of the product tester for the product of Reed-Solomon codes raises the natural question of whether the product tester may be robust for the product of any pair of codes. We note that for this question to be meaningful, it must be asked of families of codes with asymptotically good distance properties. Formally, this question takes the following form.

Question: Is the product tester always robust for general families of tensor product codes? Specifically, is it the case that there exists a function $\alpha(\delta) > 0$ such that for every infinite family of pairs of codes C_1, C_2 of relative distance δ, the tensor product $C_1 \otimes C_2$ is $\alpha(\delta)$ robust?

A somewhat more complicated tester for the tensor product of more than two codes is analyzed and shown to be robust in [5]. A positive resolution of the above question would have been sufficient for their purposes. At the same time, it would generalize the analyses of Arora and Safra [3] and Polishchuk and Spielman [10]. To date, however the status of the above question was open. Here, we provide a negative answer to this question in the form of the following theorem.

Theorem 1. *There exists an infinite family of pairs of codes $C_1 = \{C_1^{(i)}\}$ and $C_2 = \{C_2^{(i)}\}$ of relative distances δ_1, δ_2 at least $\frac{1}{10}$ such that the robustness of $C_1^{(i)} \otimes C_2^{(i)}$ converges to 0 as i increases.*

We note that it remains an interesting open problem whether the tensor product of a code with itself is always robust. Our counterexample involves the tensor product of *different* codes.

3 Proofs

As noted in the previous section, when we apply the definition of a robust local tester to the product tester of code $C = C_1 \otimes C_2$, the robustness $\rho^T(r)$ equals the average of the relative distance of r from the closest matrix with each row in C_1 and the relative distance of r from the closest matrix with each column in C_2. Defining $\delta_{C_1}(r)$ and $\delta_{C_2}(r)$ respectively to be these distances, we find the robustness of the product tester on $C = C_1 \otimes C_2$ is

$$\alpha_{C_1,C_2} = \min_r \frac{\delta_{C_1}(r) + \delta_{C_2}(r)}{2\delta_C(r)}. \tag{1}$$

Our goal then, is to find a matrix r whose rows and columns lie close to codewords of C_1 and C_2 respectively, but which lies far from any codeword of $C_1 \otimes C_2$. Specifically, we show a general construction for generating pairs of codes C_1, C_2 whose tensor product codes have arbitrarily bad robustness. This construction implies the theorem.

To motivate the following construction we note that we seek a matrix r whose rows and columns, when taken individually, are very simple, but when taken together are complicated (the rows and columns are very close to codewords of C_1 and C_2 respectively, but the whole matrix is far from any code of the tensor product). A natural matrix that fits this general rubric is the *identity* matrix. Specifically, each row or column has exactly one nonzero entry, and is thus "simple", however, when considered as a whole, the identity matrix has *full rank* which, in linear algebra terms, is as complicated as a matrix can get. We use these properties of the identity matrix in a fundamental way in the following construction.

Construction. Given an $[n, k, d]_2$ code C_1 such that the dual space $C_1^\perp$ is an $[n, n-k, d']_2$ code for some d' and an $[m, k, D]$ code C_g with the additional property that the distance of 1^m from any codeword of C_g, namely $\Delta_{C_g}(1^m)$ is also at least D, we construct a tensor product code as follows.

Let G_1 be a generator matrix for code C_1, i.e., let its rows be some basis for the subspace C_1. Similarly, let G_g be a generator for C_g. We note that G_1 will be $k \times n$ and G_g will be $k \times m$. Define the $n \times m$ matrix H to be their product $H = G_1^T G_g$.

Next, consider the $n \times mn$ matrix obtained by horizontally concatenating n copies of H, which we denote by H^n. Let I_n^m be the $n \times mn$ matrix consisting of the $n \times n$ identity matrix with each column duplicated to appear m times consecutively.

Let

$$G_2 = H^n + I_n^m$$

be the $n \times mn$ matrix that is the sum of these two matrices, and let C_2 be the code generated by the rows of G_2. Define the tensor product code $C = C_1 \otimes C_2$. □

We claim the following lemma.

Lemma 1. *The distance of the code C_2 constructed above is at least $\min\{md', nD\}$, and the robustness of the code $C = C_1 \otimes C_2$, which we denote α_{C_1,C_2}, satisfies*

$$\alpha_{C_1,C_2} \leq \frac{nm}{2(n-k)\min\{md', nD\}}. \tag{2}$$

We first show how the lemma implies our main result, and then prove the lemma.

Proof (Theorem 1). We produce a family of codes $C_1^{(i)}, C_g^{(i)}$ and show how, by Lemma 1, our construction transforms these codes into a family that satisfies the conditions of this theorem.

Let the parameters of $C_1^{(i)}$ be defined as

$$[n^{(i)}, k^{(i)}, d^{(i)}]_2 \stackrel{\text{def}}{=} [i, \frac{i}{2}, \frac{i}{10}]_2.$$

We note that from standard coding theory arguments if the $i/2 \times i$ generator matrix $G_1^{(i)}$ is filled in *at random*, then for large enough i the ensuing code will have distance $d^{(i)} \geq i/10$ with overwhelming probability. Take $C_1^{(i)}$ to be such a code. From standard linear algebra, the dual space $C_1^{\perp(i)}$ will also be a randomly generated subspace of $\mathbb{R}^i$, and thus the dual code will also have distance at least $i/10$ with overwhelming probability.

Similarly, let the $[m^{(i)}, k^{(i)}, D^{(i)}]_2$ code $C_g^{(i)}$ have parameters $[i, \frac{i}{2}, \frac{i}{10}]_2$ and be constructed from a random $i/2 \times i$ generator. Similar arguments show that for large enough i, the code C_g will also satisfy the additional property of codewords being far from 1^i with overwhelming probability.

We note that from Lemma 1, the distance of code C_2 is at least $\min\{md', nD\} = \min\{i^2/10, i^2/10\} = i^2/10$. Thus C_2 is an $[i^2, i, i^2/10]_2$ code, and the *relative* distance of C_2 is $\frac{1}{10}$, as desired.

Explicitly evaluating (2), we have that

$$\alpha_{C_1,C_2} \leq \frac{nm}{2(n-k)\min\{md', nD\}} = \frac{i^2}{2(i-\frac{i}{2})\frac{i^2}{10}} = \frac{10}{i},$$

which converges to 0 as i increases. Thus the families $C_1^{(i)}, C_2^{(i)}$ satisfy the desired properties of the theorem. □

We now prove the lemma.

Proof (Lemma 1). Consider the above construction. We first note that since both G_1 and G_g have rank k, both G_1 and G_g must have full-rank $k \times k$ submatrices. Further, the product of these submatrices will be a submatrix of their product $H = G_1^T G_g$. Thus H has rank k.

We prove now that

$$\Delta(C_2) \geq \min\{md', nD\}.$$

Consider a nonzero codeword $c \in C_2$. Since G_2 generates C_2, we have by definition that for some nonzero vector $v \in \mathbb{R}^n$, $c = v^T G_2$. We consider two cases.

In the first case, suppose $v^T H = 0$. Since H has rank k and its columns are linear combinations of rows of G_1, its column space must equal the row space of G_1, which consists of the elements of C_1. Thus $G_1 v = 0$, and we conclude that $v \in C_1^\perp$, the dual code of C_1.

Since the dual code $C_1^\perp$ has minimum distance d' by assumption, v must differ from the codeword 0^n in at least d' places. Recall that $G_2 \stackrel{\text{def}}{=} H^n + I_n^m$. Thus we have $c \stackrel{\text{def}}{=} v^T G_2 = v^T I_n^m$, from which we conclude that c differs from 0^{nm} in at least md' places.

In the second case, suppose $v^T H \neq 0$. Then since the rows of H are linear combinations of the rows of G_g, $v^T H$ is a nonzero codeword of C_g, which we denote by $x \stackrel{\text{def}}{=} v^T H$.

We note that $v^T I_n^m$ consists of n consecutive $1 \times m$ vectors that are either uniformly zero or uniformly one. Thus our codeword $c = v^T G_2 = v^T(H^n + I_n^m)$ consists of n consecutive chunks containing either x or $1-x$. From our assumptions on C_g, both x and $1-x$ differ from 0^m in at least D places. Thus c has at least nD nonzero entries.

Thus, in either case, a nonzero codeword of C_2 differs from 0^{mn} in least $\min\{md', nD\}$ places. Since codes are linear, the minimum distance of 0^{nm} to another codeword equals the minimum distance of the code, and thus $\Delta(C_2) \geq \min\{md', nD\}$, as desired.

Recall from (1) that to demonstrate the low robustness of the product tester on $C_1 \otimes C_2$ we are trying to find a matrix whose rows are very close to codes in C_1 and whose columns are very close to codes in C_2, yet which lies far from any code in the tensor product $C = C_1 \otimes C_2$. The matrix we consider here is in fact G_2^T, the transpose of the generator matrix of C_2. Note that by definition,

$$\delta_{C_2}(G_2^T) = 0,$$

since G_2 generates C_2, and thus all the columns of G_2^T are in C_2. Also, we have that

$$\delta_{C_1}(G_2^T) = 1/n,$$

since all the columns of H^n are in C_1 and each column of I_n^m has relative weight $1/n$. Thus rows and columns of G_2^T lie "close" to codewords of C_1 and C_2 respectively.

We now show that G_2^T is far from any codeword of the tensor product $C_1 \otimes C_2$, specifically, that

$$\delta_{C_1 \otimes C_2}(G_2^T) \geq \frac{(n-k)\Delta(C_2)}{n^2 m}.$$

Consider the difference $R - G_2^T$ for any codeword $R \in C_1 \otimes C_2$. We note that each column of $R - G_2^T$ is a codeword of C_2. Also, we note that each row of $R - (G_2^T - I_n^{mT})$ is a codeword of C_1, implying that $R - (G_2^T - I_n^{mT})$ has rank at most k. We now invoke the fact that the identity matrix has full rank, and conclude that the difference $[R - (G_2^T - I_n^{mT})] - I_n^{mT} = R - G_2^T$ must therefore have has rank at least $n-k$. Thus there are at least $n-k$ nonzero columns in $R - G_2^T$, each of which is a codeword of C_2. Since each nonzero codeword of C_2 has weight at least $\Delta(C_2)$, we conclude that

$$\delta_{C_1 \otimes C_2}(G_2^T) \geq \frac{(n-k)\Delta(C_2)}{n^2 m}.$$

Thus we have constructed a matrix G_2^T for which

$$\alpha_{C_1,C_2} \leq \frac{\delta_{C_1}(G_2^T) + \delta_{C_2}(G_2^T)}{2\delta_C(G_2^T)} \leq \frac{nm}{2(n-k)\Delta(C_2)},$$

as desired, and the lemma is proven. □

4 Acknowledgement

I am grateful to Madhu Sudan for introducing this problem to me, and for many valuable discussions along the way to solving it and writing it up. I would also like to acknowledge the suggestions of an anonymous referee with respect to my introduction.

References

1. Sanjeev Arora. *Probabilistic checking of proofs and the hardness of approximation problems*. PhD thesis, University of California at Berkeley, 1994.
2. Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, May 1998.

3. Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, January 1998.
4. Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of proximity, shorter PCPs and applications to coding. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 1-10, 2004.
5. E. Ben-Sasson and M. Sudan. Robust Locally Testable Codes and Products of Codes. RANDOM 2004, pages 286–297, 2004.
6. E. Ben-Sasson and M. Sudan. Simple PCPs with Poly-log Rate and Query Complexity. 37th STOC (to appear), 2005.
7. Katalin Friedl and Madhu Sudan. Some improvements to total degree tests. In *Proceedings of the 3rd Annual Israel Symposium on Theory of Computing and Systems*, pages 190–198, Tel Aviv, Israel, 4-6 January 1995. Corrected version available online at http://theory.csail.mit.edu/~madhu/papers/friedl.ps.
8. Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost-linear length. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, Vancouver, Canada, 16-19 November 2002.
9. F. J. MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes*. Elsevier/North-Holland, Amsterdam, 1981.
10. Alexander Polishchuk and Daniel A. Spielman. Nearly linear-size holographic proofs. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing*, pages 194–203, Montreal, Quebec, Canada, 23-25 May 1994.
11. Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, April 1996.
12. Ran Raz and Shmuel Safra. A Sub-Constant Error-Probability Low-Degree Test, and a Sub-Constant Error-Probability PCP Characterization of NP. In *Proceedings of the 29th STOC*, pages 475–484, 1997.
13. Madhu Sudan. Algorithmic introduction to coding theory. Lecture notes, Available from http://theory.csail.mit.edu/~madhu/FT01/, 2001.

APPENDIX

We prove here the fact mentioned in the body of the article that the product tester meets the criteria for a (non-robust) local tester.

Proof. Suppose we have a tensor product code $C = C_1 \otimes C_2$ and a matrix r for which the product tester accepts with probability α. This implies that for some $\beta + \gamma = 2\alpha$, β-fraction of the rows of r are in C_1 and γ-fraction of the columns of r are in C_2.

Consider the submatrix $r_{\beta,\gamma}$ at the intersection of these β rows and γ columns. Let G_1, G_2 be the generators of C_1, C_2 respectively. Let G_1^β and G_2^γ be the restrictions of G_1, G_2 respectively to those columns that correspond to the β rows of r and γ columns of r respectively. Since $r_{\beta,\gamma}$ is an element of the tensor product of the codes generated by G_1^β and G_2^γ, we may express $r_{\beta,\gamma}$ as

$$r_{\beta,\gamma} = G_2^{\gamma T} X G_1^\beta,$$

for some matrix X. We now extend $r_{\beta,\gamma}$ to a full matrix r', with the property that *every* row of r' is in C_1 and every column is in C_2. Specifically, let

$$r' \stackrel{\text{def}}{=} G_2^T X G_1,$$

for the full matrices G_1, G_2. Clearly r' is a codeword of C.

We note that by definition, r' agrees with r on the submatrix $r_{\beta,\gamma}$. Thus, $\delta_C(r)$, the distance of r from the nearest codeword of C is at most the distance from r to r', which is at most $1 - \beta\gamma$.

Recall that we are trying to prove that the probability of r failing the test, namely $1 - \alpha$, is at least half this distance $\delta_C(r)$. The algebra from here is straightforward. Recall that $1 - \alpha = 1 - \frac{\beta+\gamma}{2}$, and further that

$$\delta_C(r)/2 \leq \frac{1}{2} - \frac{\beta\gamma}{2} \leq \frac{1}{2} - \frac{\beta\gamma}{2} + \frac{(1-\beta)(1-\gamma)}{2} = 1 - \frac{\beta+\gamma}{2}.$$

Comparing these two equations yields the desired result.