This is the Title PhL 3.26.05

OK, I think I am done with all the stuff in this doc, and it has been incorporated into Section 1 (b) 4. This took an entire day to do. I think I have finally cleared up the vector space stuff. However, I still have to work in these facts:

GF(p) is a vector space over Z_p **DONE**

GF($q=p^m$) is a vector space over Z_p **DONE**.

But I cannot do that right in Section 1 (b) 4 because these GF objects have not yet been mentioned.
(but I threw it in there anyway)

So the symbols = elements of GF(q) themselves are elements of a vector space over field Z_p = GF(p).

But then we get to code words and things change. A data word with k components is a vector over the field GF(q). Each component can take on q values. You can add data words, etc etc. This is a different vector space concept from saying that the elements of GF(q) form a vector space over Z_p.
(I have added a comment on this at the start of Chapter 7).

Galois Fields and Vector Spaces

Is a Galois Field GF(p) a vector space? If one peruses the list of properties for a set of elements to be a vector space, one finds that GF(p) fulfills about half of these properties just because it is an abelian group under addition: associative, commutative, identity and inverse exist. The remaining properties require, in addition to the elements of GF(p), the elements of some field F. If α and β are elements of F, then we know the meaning of things like α+β and αβ just because F is a field. We know that F contains a 0 and a 1 element. If the remaining properties involving F are satisfied, then GF(p) would be a "vector field over F". If **a** and **b** are elements of GF(p) and α and β are elements of F, these extra properties are as follows:

α**a** is defined	one must clarify what it means to multiply α in F by **a** in GF(p)
α**a** = **a** for α =1	the multiplication rule above must work this way for α = 1
α(**a**+**b**) = α**a** + α**b**	distributive but α is in F while **a** and **b** are in GF(p)
(α+β)**a** = α**a** + β**a**	distributive but α and β are in F while **a** is in GF(p)
α(β**a**) = (αβ)**a**	associative but α and β are in F while **a** is in GF(p)

The field GF(p) is not a vector space over F = reals because for example (1/5)**a** is not defined. This is not an element of GF(p). GF(p) is not a vector space over the integers because the integers are a ring, not a field (inverses don't exist). GF(p) *is* a vector space over the field of integers mod p which field we called Z_p. We can go down the list above and verify this is so. For example, α**a** ≡ **a** + **a** + ...+**a** α times and since GF(p) is an abelian group under addition and is associative under addition, this sum is clearly defined. Also 1**a** = **a** seems quite clear. Then for example 2(**a**+**b**) = (**a**+**b**) + (**a**+**b**) = 2**a** + 2**b** and in this way the third property can be verified, as well as the remaining two properties.

Since GF(p) is isomorphic to Z_p, we end up in some sense with GF(p) being a field over itself, but it is best to think of the elements of GF(p) as being abstract, while Z_p is a specific field.

Question: is every additive cyclic group a vector space?

I ask this because this is where I make my first erroneous comment about vector fields. First, I have to ask: is every additive cyclic group commutative under + ? Yes, and I just added it to galois doc.

So, any additive cyclic group satisfies the first properties since it is an additive abelian group. So that part is OK, but then we come to the second set of axioms and the required field F. Would Z_n work as it does in the case of GF(p)?

(1) $\alpha\mathbf{a} \equiv \mathbf{a} + \mathbf{a} + ... + \mathbf{a}$ certainly gives a meaning $\alpha\mathbf{a}$. α could be any integer as far as this goes.

I think all the properties go through. But why not use any Z_m which is unrelated to the order n of the additive group? Well we do know that if order is n, then $n\mathbf{a} = 0$ so there is no need for any integers larger than n-1, so that is a good reason to use Z_n.

So I am ready to proclaim;

Fact: Any additive cyclic group of order n is a vector field over Z_n.

Examples: GF(p) and GF(q). So the only property of the GF(q) I need here is + cyclic!

Question: Is GF(p) a cyclic additive group?

- I show that Z/(n) = Z_n at the ring level in (1.40).
- I show that Z/(n) = Z_n is a field when n is prime in (1.44)
- I show that For p = prime, GF(p) = Z_p = Z/(p) in (2.5a)

Since Z_p is an additive cyclic group (this was the subject of Section 1 (*) 5), we conclude that GF(p) is a cyclic additive group. There must be a generator **g** such that $\mathbf{g_1} = k\mathbf{g}$ for any g_1 in the group.

Question: How do we know that *any* g is a generator in an additive cyclic group of prime order?

This is shown in (1.13) for the general * operation. So OK.

Question: Are the elements of GF(q) elements of some vector space?

First I quote,

Fact: $p\mathbf{g} = \mathbf{0}$ for any element **g** of GF(q). // compare to *(2.1)* above (2.7) (4.5)

Question: Do the elements of GF(q) form an additive cyclic group?

I would have to know that there was at least one **g** which would additively generate all p^m elements of the group, that seems unlikely to me. We know there is a • generator, but nothing has been said about +

generator. Well here is a conclusion from the Fact above. Suppose we try to enumerate all elements of GF(p^m) using some arbitrary g ≠0. We then get {**0**, **g**, 2**g**, ...p**g** = **0**, (p+1)**g** = **g** , ...} Thus, we do NOT enumerate all elements for sure, so GF(q) is not a cyclic additive group at least for m>1 since $p^m > p$ in that case.

But what about m = 1. Is it possible that GF(p) is an additive *cyclic* group? Well, in order to be cyclic, it has to have at least one generator, and **1** is such a generator! We know that 1 generates all elements of Z_p and we know that GF(p) = Z_p, so 1 must generate all elements of GF(p) as well! Therefore, GF(p) is additive cyclic. Since p is prime, all non-zero elements are generators.

Question: In my latest Section 7 opening I claim that an n-tuple is a direct sum space. Is this really correct/ If so, can I find verification? And why when you combine two spin states is it a direct sum instead of a direct product?

Here is one verification www.maths.bris.ac.uk/~maxmr/la2/notes_2.pdf

6

2.3. **Direct sums and quotients.**

Definition 2.25. Let V and W be vector spaces. Denote by $V \oplus W$ the vector space formed by all pairs (v, w) for $v \in V, w \in W$. The addition is defined as

$$(v, w) + (v', w') = (v + v', w + w').$$

The multiplication by a scalars is defined as

$$a(v, w) = (av, aw).$$

The zero vector of $V \oplus W$ is the pair $(\mathbf{0}, \mathbf{0})$.

I have also reviewed my own direct sum notes from the n-tuple point of view, it does seem exactly right. So how about the spin question? That is a direct product of group representations which my notes do talk about. I show that the direct product of two Lie Algebra representations is itself a representation. I write that as $Z_i = X_i * Y_i$ for generators. Now suppose X_i and Y_i are generator representations. Is it possible that the direct sum of these is also a representation? I would have to check. But I need some claimed formula for the generators of the direct sum space. The obvious choice would be $Z_i = X_i \oplus Y_i$. I would have to show that $[Z_i, Z_j] = c^k{}_{ij} Z_k$ if these was true for the X and Y.

$[Z_i,Z_j] = Z_iZ_j - Z_jZ_i = (X_i \oplus Y_i)(X_j \oplus Y_j) - (X_j \oplus Y_j)(X_i \oplus Y_i)$ // install obvious

$= X_iX_j \oplus Y_iY_j - [X_jX_i \oplus Y_jY_i]$ // one of my sheet rules

$= [X_iX_j - X_jX_i] \oplus [Y_iY_j - Y_jY_i]$ // this should be a rule on my sheet

$= [X_i,X_j] \oplus [Y_i,Y_j]$

$= [\Sigma_k c^k{}_{ij} X_k] \oplus [\Sigma_{k'} c^{k'}{}_{ij} Y_{k'}]$ // install separate Lie's

$= \Sigma_k c^k{}_{ij} (X_k \oplus Y_k)$ // have to do case n=2 to see this

$= \Sigma_k c^k{}_{ij} Z_i$

So as expected, the direct sum of Lie generators is a Lie generator, and then of course the direct sum of the representation matrices in block diagonal form is a group representation.

So the direct product and direct sum are both group representations, they are just different group representations! OK, I am happy with this, I don't think I am saying anything wrong.