EXPLICIT CONSTRUCTIONS OF CODE LOOPS AS CENTRALLY TWISTED PRODUCTS

TIM HSU

ABSTRACT. *Code loops* are certain Moufang loop extensions of doubly even binary codes which are useful in finite group theory (e.g., Conway's construction of the Monster). We give several methods for explicitly constructing code loops as *centrally twisted products*. More specifically, after establishing some preliminary examples, we show how to use decompositions of codes to build code loops out of more familiar pieces, such as abelian groups, extraspecial groups, or the octonion loop. In particular, we use Turyn's construction of the Golay code to give a simple explicit construction of the Parker loop, one which may have applications to the study of the Monster.

1. INTRODUCTION

The "remarkable Moufang loop" discovered by R. A. Parker first appeared in print when it played a key role in Conway's construction of the Fischer-Griess Monster group [6]. The first published proof of the existence of the Parker loop, or more generally, of the class of Moufang loops called *code loops*, was subsequently given by Griess [10], who went on to explore other constructions of nonsplit extensions using Moufang loops [11, 12, 13]. For other work arising from code loops, see, for instance, Chein and Goodaire [5], Johnson [15], and Richardson [19].

When working with a code loop $\mathcal{C}$, for many purposes, it is sufficient to use only the definition of $\mathcal{C}$ as an extension of a doubly even code (Definition 2.4, below). However, for very specific calculations, such as calculations inside the Griess algebra for the Monster (as constructed by Conway [6]), it may be necessary to have an explicit description of multiplication in $\mathcal{C}$. Such a description is given implicitly in the existence proof by Griess [10], and more explicitly by Conway and Sloane [8, Ch. 29, App. 2]; however, both of these descriptions are built up inductively over a basis, and therefore require either working with the code in terms of a basis decomposition, or constructing a lookup table. More practically, Kitazume [16] gives a description of multiplication in code loops over codes containing subcodes isomorphic to even codes over $\mathbf{F}_4$ (e.g., the hexacode inside the Golay code). However, Kitazume's construction is only useful in this circumstance; furthermore, even when the construction is applicable, an *ad hoc* step is needed to complete the construction each time, and the resulting multiplication procedure still seems somewhat involved.

In this paper, we use our previous work on code loops (or more generally, "small Frattini Moufang loops") [14] to give simple and practical explicit constructions

Date: May 4, 1998.

1991 *Mathematics Subject Classification.* Primary 20N05; Secondary 20D15, 20D08.

Key words and phrases. Code loops, Parker loop, doubly even binary codes, centrally twisted products.

for many interesting code loops. Specifically, we use the *centrally twisted product* of sub-code loops to construct code loops from smaller associative (or at least familar) pieces. We begin in Section 2 by summarizing the necessary notation, definitions, and previous results. In Section 3, we discuss the structure of some of the simpler and smaller pieces which we will use to construct larger code loops. We then use these pieces to analyze our main example, the Parker loop, in Section 4, paying special attention both to the "Turyn" structure of the Parker loop, and to issues of practical implementation. Finally, Section 5 describes a centrally twisted product decomposition which may be used to construct any code loop, though not necessarily in a convenient fashion.

Remark 1.1. By establishing these methods for explicit constructions of code loops, we hope not only to make computation more efficient in code loops and code loop-related objects (e.g., the Monster and similarly constructed sporadic groups), but also to illuminate the way in which the structure of a doubly even code is reflected in its code loop. In particular, we hope that the Turyn-type decomposition of the Parker loop provided here may eventually play a role in a Turyn-type decomposition of the Monster itself (an idea suggested by a lecture of S. Smith).

2. Doubly even codes, code loops, and centrally twisted products

Notation. In this paper, we use the usual additive notation for finite fields and vector spaces, instead of the multiplicative notation often used in finite group theory papers (e.g., in Conway [6] or in [14]). Let V and W be subspaces of $\mathbf{F}_p^n$. We use $V + W$ to denote the space spanned by V and W, and call this the *sum* of V and W, even when V and W are linearly independent. We reserve the symbol $V \oplus W$, and the term *direct sum*, for the case when V and W have no nonzero coordinates in common.

We consider a vector $v = (v_i) \in \mathbf{F}_2^n$ to be a subset of $\{1, \ldots, n\}$ by associating v with the set of all $i \in \{1, \ldots, n\}$ such that $v_i = 1$. In this notation, $u + v$ is the symmetric difference of the associated sets, and $|u|$ (resp. $|u \cap v|$, $|u \cap v \cap w|$) is the number of coordinates which are nonzero in u (resp. both u and v, all of u, v, and w). $|u|$ is called the *weight* of u.

We write the group of order 2 multiplicatively, with elements $\{\pm 1\}$.

If $a, b, c, \ldots$ are elements of a vector space, loop, or group, $\langle a, b, c, \ldots \rangle$ denotes the object generated by $a, b, c, \ldots$.

(n, s) will sometimes be used to denote the greatest common divisor of the integers n and s.

We recall the basic definitions of code loops (Definitions 2.1–2.4).

Definition 2.1. A *doubly even binary code* of length n and dimension k is a subspace C of dimension k in $\mathbf{F}_2^n$ such that $|c| \equiv 0 \pmod 4$ for all $c \in C$. The vectors of C are also called *codewords*. If a nonzero codeword of C with smallest possible weight has weight d, we say that C is an $[n, k, d]$ code.

Let Z be the group of order 2. Given any doubly even code C, we define the functions $\sigma : C \to Z$, $\chi : C \times C \to Z$, and $\alpha : C \times C \times C \to Z$ by

$$\sigma(c) = (-1)^{|c|/4} \tag{2.1}$$

$$\chi(c, d) = (-1)^{|c \cap d|/2} \tag{2.2}$$

$$\alpha(c, d, e) = (-1)^{|c \cap d \cap e|}, \tag{2.3}$$

for all $c, d, e \in C$.

Clearly χ and α are symmetric in all variables. Also, χ and α are symplectic; that is,

$$\chi(c, c) = +1 \tag{2.4}$$

$$\alpha(c, c, d) = +1. \tag{2.5}$$

Furthermore, since $|c + d| = |c| + |d| - 2\,|c \cap d|$,

$$\sigma(c + d) = \sigma(c)\sigma(d)\chi(c, d) \tag{2.6}$$

$$\chi(c + d, e) = \chi(c, e)\chi(d, e)\alpha(c, d, e) \tag{2.7}$$

$$\alpha(c + d, e, f) = \alpha(c, e, f)\alpha(d, e, f). \tag{2.8}$$

In other words, σ, χ, and α are related by polarization.

Definition 2.2. A *loop* is a set L with a binary operation (written as juxtaposition) and an element $1 \in L$ such that for all $a \in L$, the maps $x \mapsto xa$ and $x \mapsto ax$ are bijections from L to itself, and $1a = a1 = a$. A *Moufang loop* is a loop L such that $(xy)(zx) = x((yz)x)$ for all $x, y, z \in L$.

We define Moufang loops only for the sake of completeness, since we will not use any of their properties directly. For basic facts about (Moufang) loops, see Bruck [4] and Pflugfelder [18].

Definition 2.3. Let L be a loop. For $x, y, z \in L$, the commutator (resp. associator) $[x, y]$ (resp. $[x, y, z]$) is defined to be the element of L such that $xy = (yx)[x, y]$ (resp. $(xy)z = (x(yz))[x, y, z]$).

We use the definition of code loop from Griess [11].

Definition 2.4. Let C be a doubly even code. We say that a loop $\mathcal{C}$ is a *code loop* of C if:

1. $\mathcal{C}$ has a central subgroup Z (i.e., a subloop $Z \leq \mathcal{C}$ such that $[z, \gamma] = [z, \gamma, \delta] = 1$ for all $z \in Z$, $\gamma, \delta \in \mathcal{C}$) of order 2 such that $\mathcal{C}/Z \cong C$, as an elementary abelian 2-group. (Note that if Φ is the resulting natural homomorphism from $\mathcal{C}$ to C, $\Phi(\gamma) = c$, and $\Phi(\delta) = d$, then $\Phi(\gamma\delta) = c + d$.)
2. For all $\gamma, \delta, \epsilon \in \mathcal{C}$ with images $c, d, e \in C \bmod Z$, we have

$$\gamma^2 = \sigma(c) \tag{2.9}$$

$$[\gamma, \delta] = \chi(c, d) \tag{2.10}$$

$$[\gamma, \delta, \epsilon] = \alpha(c, d, e). \tag{2.11}$$

Note that (2.11) allows us to associate terms in a code loop as long as we apply the correct error term of ± 1. We leave it as an exercise for the reader to use this idea and the (skew-)symmetric and multilinear properties of α to verify that code loops are Moufang. Many other identities may be similarly computed; for instance, we have

$$\gamma(\delta\epsilon) = \delta(\gamma\epsilon) \cdot \chi(c, d) \qquad \text{and} \tag{2.12}$$

$$(\gamma\delta)\epsilon = (\gamma\epsilon)\delta \cdot \chi(d, e). \tag{2.13}$$

We come to the following result of Griess [10] (see also [14]).

Theorem 2.5. *The code loop of a doubly even code exists and is unique up to isomorphism.* □

We next quote Definition 2.6 and Theorem 2.7 from [14], in a customized form. (Note that similar constructions appear in Johnson [15] and Kitazume [16].)

Definition 2.6. Let D and E be subcodes of a doubly even code C, and let $\mathcal{D}$ and $\mathcal{E}$ be the code loops of D and E, respectively. The binary operation on the set $\mathcal{D} \times \mathcal{E}$ given by

$$(\delta_1, \epsilon_1)(\delta_2, \epsilon_2) = (\chi(e_1, d_2)\alpha(d_1, e_1 + d_2, e_2)\delta_1\delta_2, \epsilon_1\epsilon_2), \tag{2.14}$$

defines a loop Γ containing a central subgroup $Z \times Z$. We define the *centrally twisted product* of $\mathcal{D}$ and $\mathcal{E}$ to be $\Gamma / \langle (-1,-1) \rangle$. We denote the centrally twisted product of $\mathcal{D}$ and $\mathcal{E}$ by $\mathcal{D} \circledast \mathcal{E}$.

Note that if $C = D \oplus E$, or more generally, if the "twisting term" involving χ and α vanishes, then the centrally twisted product $\mathcal{D} \circledast \mathcal{E}$ becomes the central product $\mathcal{D} \bigcirc \mathcal{E}$.

Theorem 2.7. *Let D and E be linearly independent subcodes of a doubly even code C, and let $\mathcal{D}$ and $\mathcal{E}$ be the code loops of D and E. Then $\mathcal{D} \circledast \mathcal{E}$ is the code loop of $D + E$.* □

As we will see in Section 4, the point of Theorem 2.7 is that the definitions of χ and α imply that multiplication in the code loop of $D + E$ must actually be described by (2.14). Theorem 2.7 is therefore really just a matter of verifying that the product defined by (2.14) satisfies (2.9)–(2.11). See [14] for details.

3. Some easy examples and small examples

We first consider some easy examples and some small examples of doubly even codes and their code loops.

Example 3.1. If C is a doubly even code of dimension k such that $\sigma = +1$ identically, then (2.6) and (2.7) imply that $\chi = \alpha = +1$ identically, and so the code loop of C is an elementary abelian 2-group of order 2^{1+k}. We call such a code loop *split*. Note that if C is *any* doubly even code, then the "double" of C (i.e., the subcode $\{(c,c) \mid c \in C\}$ of $C \oplus C$) is split.

Example 3.2. The doubly even code d_{2n} ($n \geq 2$) may be defined to be the code with basis $\langle c_2, \ldots, c_n \rangle$, where the c_i are the following vectors.

$$\begin{array}{lcccccccccccccc}
c_2 & 1 & 1 & 1 & 1 & & & & & & & & & & \\
c_3 & 1 & 1 & & & 1 & 1 & & & & & & & & \\
c_4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & & & & \\
c_5 & 1 & 1 & 1 & 1 & 1 & 1 & & & 1 & 1 & & & & \\
c_6 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & \\
c_7 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & 1 & 1 \\
\vdots & & & & & & & \vdots & & & & & & & \\
\end{array} \tag{3.1}$$

Let $\mathcal{D}_{2n}$ be the code loop of d_{2n}. Recall (Aschbacher [1]) that the *extraspecial group* 2_+^{1+2k} (resp. 2_-^{1+2k}) is the central product of k copies of the dihedral group of order 8 (resp. $k-1$ copies of the dihedral group of order 8 and 1 copy of the quaternion group of order 8). It follows from inspection of our chosen basis that $\mathcal{D}_{2n}$ is the extraspecial group $2_+^{1+(n-1)}$ (resp. $2_-^{1+(n-1)}$) for $n \equiv \pm 1$ (resp. ± 3) mod 8, and that for $n \equiv 0$ (resp. 2) mod 4, $\mathcal{D}_{2n}$ is the central product of a Klein four-group (resp. cyclic group of order 4) and $\mathcal{D}_{2n-2}$.

Example 3.3. To describe the Hamming codes we need, we use coordinates compatible with the MOG (see Conway and Sloane [8, Ch. 11]). Specifically, we write $\mathbf{F}_2^8$ in a 4×2 array, which we call an *octad*, representing 0 and 1 by blank and non-blank symbols, respectively.

We define the *Hamming point-code* (resp. *Hamming line-code*) $e_7(P)$ (resp. $e_7(L)$) to be the code consisting of the codewords in row (P) (resp. row (L)) of Figure 1, and we define $e_8(P)$ (resp. $e_8(L)$) to be the code consisting of the codewords in $e_7(P)$ (resp. $e_7(L)$) and their complements.

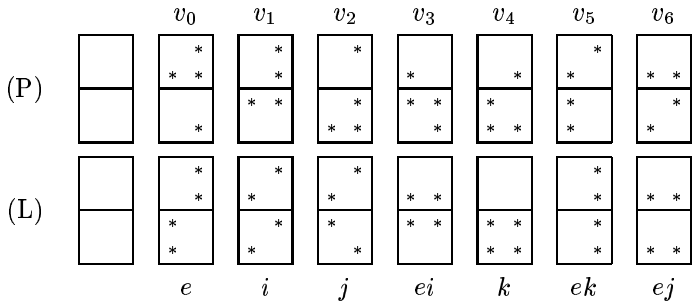

FIGURE 1. The point-code and the line-code

Let $\mathcal{E}_7$ (resp. $\mathcal{E}_8$) be the code loop of e_7 (resp. e_8), in either the point or line versions. Since e_8 is just e_7 with the all-ones codeword adjoined, $\mathcal{E}_8 \cong 2 \times \mathcal{E}_7$, so it will suffice to discuss multiplication in $\mathcal{E}_7$, which we will do in some detail, as it will be quite useful to develop computational facility with $\mathcal{E}_7$ and $\mathcal{E}_8$.

Now, the vectors v_0, v_1, and v_2 in Figure 1 are a basis for e_7, so if we (arbitrarily) choose preimages $e, i, j \in \mathcal{E}_7$ of v_0, v_1, v_2, Theorem 2.7 then implies that $\mathcal{E}_7 = \langle e \rangle \oplus (\langle i \rangle \oplus \langle j \rangle)$. By collecting signs in front, we may therefore represent every element of $\mathcal{E}_7$ uniquely in the form $\pm(e^p(i^q j^r))$, where p, q, and r are either 0 or 1. Note that if $\gamma = \pm(e^p(i^q j^r))$, then p, q, and r can be read from the codeword $\overline{\gamma}$, as shown in Figure 2.

FIGURE 2. Coordinates in the point-code and line-code

Multiplication in $\mathcal{E}_7$ then becomes

$$(e^{p_1}(i^{q_1} j^{r_1}))(e^{p_2}(i^{q_2} j^{r_2})) = (-1)^s (e^{p_1+p_2}(i^{q_1+q_2} j^{r_1+r_2})), \tag{3.2}$$

where

$$s = p_1 q_1 r_2 + p_1 r_1 q_2 + q_1 p_2 + r_1 p_2 + r_1 q_2, \tag{3.3}$$

and $-1 = e^2 = i^2 = j^2$. Setting $k = ij$, and checking the resulting multiplication table, we see that $\mathcal{E}_7$ is precisely the *octonion loop of order 16*. (For example, compare the table in Pflugfelder [18, IV.1.1].)

The easiest method for computing in the octonion loop may well be the following. For $r \in \mathbf{Z}/7$, let i_r be the octonion indicated in the v_r column of Figure 1 (e.g., $i_3 = ei$). Then for all $r \in \mathbf{Z}/7$, we have $\langle i_r, i_{r+1}, i_{r+3} \rangle$ associative, and

$$-1 = i_r^2 = i_r i_{r+1} i_{r+3}. \tag{3.4}$$

For instance, $i_4 i_5 = i_0$ and $i_6 i_2 = -i_0$. See Coxeter [9] for more about this description of the octonions, including its relation to the E_8 lattice and the projective plane over $\mathbf{F}_2$.

Incidentally, the numbering $v_0 \dots v_6$ used in Figure 1 may be obtained in the following manner. To each $c \in e_7(P)$ (resp. $e_7(L)$) with coordinates p, q, and r from Figure 2(P) (resp. Figure 2(L)), we associate the element $p + qx + rx^2$ of $\mathbf{F}_2[x]/(x^3+x+1) \cong \mathbf{F}_8$. Then the codeword v_m is the one whose associated element of $\mathbf{F}_8$ is equal to x^m.

We remark that Examples 3.2 and 3.3 allow us to calculate in a large class of examples of code loops because of the following theorem of Conway and Pless [7].

Theorem 3.4. *Any doubly even code generated by its tetrads (codewords of weight 4) is the direct sum of copies of d_{2n}, e_7, and e_8.* □

It follows that the code loop of a doubly even code generated by its tetrads is the central product of copies of $\mathcal{D}_{2n}$, $\mathcal{E}_7$, and $\mathcal{E}_8$.

4. The Parker loop

In this section, as a demonstration of the use of centrally twisted products, and as a goal of independent interest, we obtain a simple but explicit description of the Parker loop. We begin with Turyn's construction of the Golay code (see Conway and Sloane [8, Ch. 11]).

Definition 4.1. Writing $\mathbf{F}_2^{24}$ as three octad "coordinates" (i.e., using the MOG, cf. Example 3.3), we define

$$A = \{(\overline{a}, 0, \overline{a}) \mid \overline{a} \in e_8(L)\} \tag{4.1}$$

$$B = \{(0, \overline{b}, \overline{b}) \mid \overline{b} \in e_8(L)\} \tag{4.2}$$

$$T = \{(\overline{t}, \overline{t}, \overline{t}) \mid \overline{t} \in e_8(P)\}. \tag{4.3}$$

We define the *Golay code* $\mathcal{G}$ to be the (doubly even) code $A + B + T$.

Let $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{T}$ be the code loops of A, B, and T. If we define $\mathcal{P}$ (the *Parker loop*) to be the code loop of $\mathcal{G}$, appplying Theorem 2.7 twice, we see that $\mathcal{P} \cong (\mathcal{A} \oplus \mathcal{B}) \oplus \mathcal{T}$. More explicitly, using the fact that $\mathcal{A}$ and $\mathcal{B}$ are split code loops (Example 3.1), and collecting signs in the third coordinate, we may represent elements of $\mathcal{P}$ as triples (a, b, τ) ($a \in A$, $b \in B$, $\tau \in \mathcal{T}$), with multiplication given by

$$(a_1, b_1, \tau_1)(a_2, b_2, \tau_2) = (a_1 + a_2, b_1 + b_2, z\tau_1\tau_2) \tag{4.4}$$

where

$$z = \chi(t_1, a_2 b_2)\alpha(a_1 b_1, t_1 a_2 b_2, t_2)\chi(b_1, a_2)\alpha(a_1, b_1 a_2, b_2) \tag{4.5}$$

and t_i is the homomorphic image of τ_i in T. Note that the computation of (4.4) relies on being able to multiply in $\mathcal{T}$, so (4.4) is not quite as simple as it may seem. However, since $\mathcal{T} \cong \mathcal{E}_8(P)$, multiplication in $\mathcal{T}$ may be handled with any of the methods of Example 3.3, and therefore provides relatively little computational difficulty.

Another drawback of the above representation is that we need 73 binary "bits" to represent an element of $\mathcal{P}$. However, since A and B are doubled codes, and T is a tripled code, if we replace (a, b, τ) with the corresponding $(\overline{a}, \overline{b}, \overline{\tau})$ ($\overline{a}, \overline{b} \in e_8(L)$ and $\overline{\tau} \in \mathcal{E}_8(P)$), we lose no information. For instance, in this reduced notation, multiplication $\mathcal{P}$ will still be given by

$$(\overline{a}_1, \overline{b}_1, \overline{\tau}_1)(\overline{a}_2, \overline{b}_2, \overline{\tau}_2) = (\overline{a}_1 + \overline{a}_2, \overline{b}_1 + \overline{b}_2, \overline{z}\,\overline{\tau}_1\overline{\tau}_2). \tag{4.6}$$

for some $\overline{z} = \pm 1$.

Now, as before, $\overline{z}$ is determined by values of χ and α on the doubled and tripled codes A, B, and T, so to compute $\overline{z}$, we could just double or triple $\overline{a}_i$, $\overline{b}_i$, and $\overline{\tau}_i$, as appropriate, and compute χ and α in the Golay code. However, more efficiently, we may actually calculate the values of χ and α as if $\overline{a}_i$, $\overline{b}_i$, and $\overline{t}_i$ were all contained within the same octad, as long as we "square" the values of χ and α when appropriate. That is, let χ_0 and α_0 be the functions χ and α computed inside a single octad, and define $\overline{\chi}$ by

$$\overline{\chi}(\overline{a}_1, \overline{a}_2) = \overline{\chi}(\overline{b}_1, \overline{b}_2) = +1 \tag{4.7}$$

$$\overline{\chi}(\overline{t}_1, \overline{t}_2) = \chi_0(\overline{t}_1, \overline{t}_2) \qquad \overline{\chi}(\overline{a}, \overline{b}) = \chi_0(\overline{a}, \overline{b}) \tag{4.8}$$

$$\overline{\chi}(\overline{a}, \overline{t}) = (-1)^{|\overline{a} \cap \overline{t}|} \qquad \overline{\chi}(\overline{b}, \overline{t}) = (-1)^{|\overline{b} \cap \overline{t}|} \tag{4.9}$$

and $\overline{\alpha}$ by

$$\overline{\alpha}(\overline{a}_1, \overline{a}_2, \overline{a}_3) = \overline{\alpha}(\overline{b}_1, \overline{b}_2, \overline{b}_3) = +1 \tag{4.10}$$

$$\overline{\alpha}(\overline{t}_1, \overline{t}_2, \overline{t}_3) = \alpha_0(\overline{t}_1, \overline{t}_2, \overline{t}_3) \tag{4.11}$$

$$\overline{\alpha}(\overline{a}_1, \overline{b}_1, \overline{b}_2) = \alpha_0(\overline{a}_1, \overline{b}_1, \overline{b}_2) \qquad \overline{\alpha}(\overline{a}_1, \overline{a}_2, \overline{b}_1) = \alpha_0(\overline{a}_1, \overline{a}_2, \overline{b}_1) \tag{4.12}$$

$$\overline{\alpha}(\overline{a}_1, \overline{b}_1, \overline{t}_1) = \alpha_0(\overline{a}_1, \overline{b}_1, \overline{t}_1) \tag{4.13}$$

$$\overline{\alpha}(\overline{a}_1, \overline{a}_2, \overline{t}_1) = \overline{\alpha}(\overline{a}_1, \overline{t}_1, \overline{t}_2) = \overline{\alpha}(\overline{b}_1, \overline{b}_2, \overline{t}_1) = \overline{\alpha}(\overline{b}_1, \overline{t}_1, \overline{t}_2) = +1 \tag{4.14}$$

for all $\overline{a}_i, \overline{b}_i \in e_8(L)$ and $\overline{t}_i \in e_8(P)$, in the cases listed and their symmetrized versions. Then, using (2.7)–(2.8) and (4.7)–(4.14) to expand (4.5), we have

$$\begin{aligned} \overline{z} = {} & \overline{\alpha}(\overline{t}_1, \overline{a}_2, \overline{b}_2)\overline{\alpha}(\overline{a}_1, \overline{b}_2, \overline{t}_2)\overline{\alpha}(\overline{b}_1, \overline{a}_2, \overline{t}_2)\overline{\alpha}(\overline{a}_1, \overline{b}_1, \overline{b}_2)\overline{\alpha}(\overline{a}_1, \overline{a}_2, \overline{b}_2) \\ & \cdot \overline{\chi}(\overline{t}_1, \overline{a}_2)\overline{\chi}(\overline{t}_1, \overline{b}_2)\overline{\chi}(\overline{b}_1, \overline{a}_2), \end{aligned} \tag{4.15}$$

where $\overline{t}_i$ is the homomorphic image of $\overline{\tau}_i$ in $e_8(L)$.

For the purposes of hand calculation, it may be more convenient to think of elements of $\mathcal{P}$ as "words" in generators $\overline{a}$, $\overline{b}$, and $\pm\overline{t}$ ($\overline{a}, \overline{b} \in e_8(L)$, $\overline{t} \in e_8(P)$), multiplied together by the following rules:

1. Two generators of type $\overline{a}$ (resp. type $\overline{b}$) are multiplied together by addition of code words.
2. Two generators of type $\pm\overline{t}$ are multplied as if they were elements of $\mathcal{E}_8(P)$, collecting signs in the front.
3. Elements commute and associate as described by (4.7)–(4.14).

The word problem for this loop may then be solved by the normal form $\pm\left(\overline{a}\overline{b}\right)\overline{t}$.

Example 4.2. To illustrate our construction, we consider an example, using the "generators and relations" version of the Parker loop described above. Let a_i, b_i, and t_i ($i = 1, 2, 3$) be the Parker loop elements shown in Figure 3. (The sign $+$ is assumed for the t_i's.) Following the "word problem" procedure above, we may

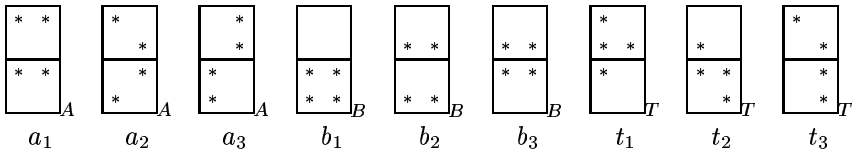

FIGURE 3. Some elements of the Parker loop

calculate, for instance:

$$\begin{aligned}
(a_1b_1)(((b_2t_1)a_2)t_2) &= +(a_1b_1)(((b_2a_2)t_1)t_2), && \text{since } \overline{\chi}(t_1,a_2)=+1,\\
&= +(a_1b_1)((b_2a_2)(t_1t_2)), && \text{from (4.14)},\\
&= +(a_1b_1)((b_2a_2)t_3), && \text{since } i_2i_3=+i_5,\\
&= -((a_1b_1)(b_2a_2))t_3, && \text{since } \overline{\alpha}(a_1,b_2,t_3)\overline{\alpha}(b_1,a_2,t_3)=-1,\\
&= +(a_1(b_1(b_2a_2)))t_3, && \text{since } \overline{\alpha}(a_1,b_1,b_2)\overline{\alpha}(a_1,b_1,a_2)=-1,\\
&= -(a_1((b_1b_2)a_2))t_3, && \text{since } \overline{\alpha}(b_1,b_2,a_2)=-1,\\
&= -(a_1(b_3a_2))t_3, && \text{since } b_1+b_2=b_3,\\
&= +(a_1(a_2b_3))t_3, && \text{since } \overline{\chi}(b_3,a_2)=-1,\\
&= -((a_1a_2)b_3)t_3, && \text{since } \overline{\alpha}(a_1,a_2,b_3)=-1,\\
&= -(a_3b_3)t_3, && \text{since } a_1+a_2=a_3.
\end{aligned}$$

Note that the $i_2i_3=+i_5$ in the third line of (4.2) refers to the version of octonion multiplication described in Example 3.3 and Figure 1.

We remark that the above example is made slightly easier by the fact that we are not finding the product of two elements already in normal form, and so we do not need to rearrange the terms as much as we otherwise might have to. This points out the fact that it is generally more efficient to wait until the very last step of a calculation to put elements in normal form.

Remark 4.3. Compare the explicit description of multiplication in the Parker loop given by Kitazume [16], who first describes the code loop over a code isomorphic to the hexacode, then describes a code loop over a complement, and then combines the two using what is essentially a special case of the centrally twisted product.

5. A GENERIC DECOMPOSITION METHOD

Since we were able to give a natural explicit construction of the Parker loop using a particular decomposition of the Golay code, the question arises: can we do the same thing for any code loop? In this section, we give a qualified answer of "yes," the catch being that the decomposition we describe is unnatural enough that it may not be much more useful for direct calculation than a basis decomposition is. However, presenting these results still seems worthwhile, since we believe they give a good indication of what the best possible results usually look like.

Let C be a doubly even binary code with code loop $\mathcal{C}$. We say that a subcode C_0 of C is *α-isotropic* if $\alpha=+1$ identically on C_0. Our goal is to express C as a sum of α-isotropic subcodes, for in terms of code loops, this expresses $\mathcal{C}$ as a centrally twisted product of groups (the code loops of the subcodes). The basic result is Theorem 5.2, the statement of which uses the following function.

Definition 5.1. For any positive k, we define $\mu(k)$ to be the largest nonnegative integer m such that $\dfrac{m(m+1)}{2} < k$.

For instance, $\mu(1) = 0$, $\mu(2) = \mu(3) = 1$, $\mu(4) = \mu(5) = \mu(6) = 2$, $\mu(k) = 3$ for $7 \le k \le 10$, and so on.

Theorem 5.2. *Let C be a doubly even binary code of dimension k. Then C contains an α-isotropic subcode C_0 of dimension $\mu(k)+1$.*

Proof. Since any subcode of dimension ≤ 2 is α-isotropic, by induction, it is enough to show that any α-isotropic subcode C_0 of dimension $m = \dim C_0 \le \mu(k)$ is contained in some α-isotropic subcode C_1 of dimension $m+1$. Let D be an arbitrarily chosen complement of C_0, and let $A = \bigwedge^2 C_0$, the alternating square of C_0. Since α is symplectic, by the universal properties of the alternating square (see Lang [17, Ch. XVI]), α determines a linear map Φ from D to A^*, the dual space of A. However, since $m \le \mu(k)$, $k > \dfrac{m(m+1)}{2}$, which means that $\dim D = k - m > \dfrac{m(m-1)}{2} = \dim A^*$. Therefore, there exists some $d \neq 0$ in $\ker \Phi$, i.e., some nonzero $d \in D$ such that $\alpha(d, c_1, c_2) = +1$ for all $c_1, c_2 \in C_0$. $C_1 = \langle C_0, d \rangle$ is therefore an α-isotropic subcode of dimension $m+1$, and the theorem follows. □

Remark 5.3. Note that the bilinear analogue of Theorem 5.2 is that any vector space C over $\mathbf{F}_2$ of dimension k with an attached symplectic bilinear form has an isotropic subspace of dimension $\mu + 1$, where μ is the greatest integer strictly less than $n/2$. It therefore seems reasonable to suspect that, when α is nondegenerate, the subcode C_0 in Theorem 5.2 will generally be the largest possible such subcode.

The point of Theorem 5.2 is that by applying it repeatedly, we are guaranteed to get a decomposition of C as the sum of α-isotropic subcodes. For example, suppose $\dim C = 12$. Applying Theorem 5.2 to C gives an α-isotropic subcode C_0 of dimension 5 such that $C = C_0 + D$ for some subcode D of dimension 7, and applying Theorem 5.2 to D then gives an α-isotropic subcode C_1 of dimension 4 such that $D = C_1 + E$ for some subcode E of dimension 3. E may then be decomposed into subcodes of dimensions 2 and 1, both of which are α-isotropic, and so we obtain a decomposition of C into α-isotropic subcodes of dimensions $5, 4, 2, 1$. To further illustrate our result, in Table 1, we have listed the dimensions of the α-isotropic subcode decomposition of a code C guaranteed by Theorem 5.2 for $1 \le \dim C \le 16$.

$\dim C$		$\dim C$		$\dim C$		$\dim C$	
1	1	5	3,2	9	4,3,2	13	5,4,3,1
2	2	6	3,2,1	10	4,3,2,1	14	5,4,3,2
3	2,1	7	4,2,1	11	5,3,2,1	15	5,4,3,2,1
4	3,1	8	4,3,1	12	5,4,2,1	16	6,4,3,2,1

TABLE 1. α-isotropic decompositions guaranteed by Theorem 5.2

Remark 5.4. Note that in Section 4, if we adjoin the all 1's codeword to the subcode A, replace T with its subcode T_0 projecting to $e_7(P)$, and then (arbitrarily)

express T_0 as the sum of subcodes of dimension 2 and 1, we obtain a $5, 4, 2, 1$ α-isotropic decomposition of the Golay code, matching the decomposition guaranteed in Table 1.

Remark 5.5. Since the material in this section really only relies on the symplectic trilinear form α, and no reference is made to characteristic 2, we may replace the code C in Theorem 5.2 with a *symplectic cubic space* C. Our results therefore also give decompositions of small Frattini Moufang loops as centrally twisted products of groups. For more on small Frattini Moufang loops and symplectic cubic spaces, see [14]; for more on trilinear forms and commutative Moufang loops, see Bénéteau [2, 3].

6. Acknowledgements

The author was partly supported by a University of Michigan Rackham Summer Faculty Fellowship, and by the generosity of MSRI and its staff. (Research at MSRI is supported in part by NSF grant DMS-9022140.) The author would also like to thank J. H. Conway, G. Glauberman, and R. L. Griess for their help.

References

1. M. Aschbacher, *Finite group theory*, Cambridge Univ. Press, 1986.
2. L. Bénéteau, *Commutative Moufang loops and related groupoids*, Quasigroups and Loops: Theory and Applications (O. Chein, H. O. Pflugfelder, and J. D. H. Smith, eds.), Sigma Series in Pure Mathematics, vol. 8, Heldermann Verlag, Berlin, 1990, pp. 115–142.
3. ———, *The symplectic trilinear mappings: an algorithmic approach of the classification; case of the field $GF(3)$*, Applied algebra, algebraic algorithms and error-correcting codes (Tokyo, 1990), Lect. Notes Comp. Sci., vol. 508, Springer-Verlag, Berlin, 1991, pp. 267–269.
4. R. H. Bruck, *A survey of binary systems*, Ergebnisse der Mathematik und ihrer Grenzgebiete, vol. 20, Springer-Verlag, New York, 1971.
5. O. Chein and E. Goodaire, *Moufang loops with a unique nonidentity commutator (associator, square)*, J. Alg. **130** (1990), 369–384.
6. J. H. Conway, *A simple construction for the Fischer-Griess monster group*, Invent. Math. **79** (1985), 513–540.
7. J. H. Conway and V. Pless, *On the enumeration of self-dual codes*, J. Comb. Thy. A **28** (1980), 26–53.
8. J. H. Conway and N. J. A. Sloane, *Sphere packings, lattices, and groups*, 2nd ed., Springer-Verlag, New York, 1993.
9. H. S. M. Coxeter, *Integral Cayley numbers*, Duke J. Math. **13** (1946), 561–578.
10. R. L. Griess, *Code loops*, J. Alg. **100** (1986), 224–234.
11. ———, *Sporadic groups, code loops, and nonvanishing cohomology*, J. Pure Appl. Alg. **44** (1987), 191–214.
12. ———, *Code loops and a large finite group containing triality for D_4*, Rend. Circ. Mat. Palermo (2) Suppl. (1988), no. 19, 79–98.
13. ———, *A Moufang loop, the exceptional Jordan algebra, and a cubic form in 27 variables*, J. Alg. **131** (1990), 281–293.
14. T. Hsu, *Class 2 and small Frattini Moufang loops*, submitted.
15. P. M. Johnson, *Loops of nilpotence class two*, to appear in J. Alg.
16. M. Kitazume, *Code loops and even codes over F_4*, J. Alg. **118** (1988), 140–149.
17. S. Lang, *Algebra*, 2nd ed., Addison-Wesley, Menlo Park, CA, 1984.
18. H. O. Pflugfelder, *Quasigroups and loops: Introduction*, Sigma Series in Pure Mathematics, vol. 7, Heldermann Verlag, Berlin, 1990.
19. T. M. Richardson, *Local subgroups of the Monster and odd code loops*, Trans. AMS **347** (1995), no. 5, 1453–1531.

Department of Mathematics, University of Michigan, Ann Arbor, MI 48109
E-mail address: timhsu@math.lsa.umich.edu