Games of No Chance
MSRI Publications
Volume **29**, 1996

Error-Correcting Codes Derived from Combinatorial Games

AVIEZRI S. FRAENKEL

Abstract. The losing positions of certain combinatorial games constitute linear error-detecting and -correcting codes. We show that a large class of games, which can be cast in the form of *annihilation games*, provides a potentially polynomial method for computing codes (*anncodes*). We also give a short proof of the basic properties of the previously known *lexicodes*, which were defined by means of an exponential algorithm, and are related to game theory. The set of lexicodes is seen to constitute a subset of the set of anncodes. In the final section we indicate, by means of an example, how the method of producing lexicodes can be applied optimally to find anncodes. Some extensions are indicated.

1. Introduction

Connections between combinatorial games (simply *games* in the sequel) and linear error-correcting codes (*codes* in the sequel) have been established in [Conway and Sloane 1986; Conway 1990; Brualdi and Pless 1993], where lexicodes, and some of their connections to games, are explored. Our aim is to extend the connection between games and codes to a large class of games, and to formulate a potentially polynomial method for generating codes from games. We also establish the basic properties of lexicodes by a simple, transparent method.

Let Γ, any finite digraph, be the *groundgraph* on which we play the following general two-player game. Initially, distribute a positive finite number of tokens on the vertices of Γ. Multiple occupation is permitted. A move consists of selecting an occupied vertex and moving a single token from it to a neighboring vertex, occupied or not, along a directed edge. The player first unable to move loses and the opponent wins. If there is no last move, the play is declared a draw. It is easy to see (since Γ is finite) that a draw can arise only if Γ is *cyclic*, that is, Γ has cycles or loops. Games in this class—which includes Nim and Nim-like games for the case where Γ is acyclic—have polynomial strategies, in general [Fraenkel $\geq$ 1997]. It turns out that the P-positions (positions from which the player who just moved has a winning strategy) of any game in this class constitute a code.

It further turns out that, if Γ is cyclic, the structure of the P-positions is much richer if the above described game is replaced by an *annihilation game* (*anngame* for short). In such a game, when a token is moved onto a vertex u, the number of tokens on u is reduced modulo 2. Thus there is at most one token at any vertex, and when a token is moved to a vertex occupied by another, both are removed from the game.

If Γ is acyclic, it is easy to see by game-strategy considerations (or using the Sprague–Grundy function defined in Section 3) that the strategies of a non-annihilation game and the corresponding anngame are identical, so both have the same P-positions—only the length of play may be affected. Thus, for the prospect of constructing efficient codes and for the sake of a unified treatment, we may as well assume that all our games are anngames.

Summarizing, we can, without loss of generality, concentrate on the class of anngames. An anngame is defined by its groundgraph Γ, a finite digraph. There is an initial distribution of tokens, at most one per vertex. A move consists of selecting an occupied vertex and moving its token to a neighboring vertex u along a directed edge. If u was occupied prior to this move, the incoming and resident tokens on u are both annihilated (disappear from play). The player first unable to move loses and the opponent wins. If there is no last move, the outcome is a draw.

With an anngame A played on a groundgraph Γ, we associate its *annihilation graph* $G = (V, E)$, or *anngraph* for short, as follows. The vertex set V is the set of positions of A, and for $u, v \in V$ there is an edge $(u, v) \in E$ if and only if there is a move from u to v in A. We review the following basic facts, which can be found in [Fraenkel 1974; Fraenkel and Yesha 1976; 1979; 1982] (especially the latter), [Yesha 1978; Fraenkel, Tassa and Yesha 1978].

Like any finite digraph, G has a *generalized Sprague–Grundy function* γ. This function was first defined in [Smith 1966], and later expounded in [Fraenkel and Perl 1975]. See [Fraenkel 1996, p. 20] in this volume for its definition, and [Fraenkel and Yesha 1986] for full details. Let $V^f \subset V$ be the set of vertices on which γ is finite. If we make V into a vector space over GF(2) in the obvious way, then V^f is a linear subspace, and γ is a homomorphism from V^f onto $\mathrm{GF}(2)^t$, for some $t \in \mathbb{Z}^0 := \{k \in \mathbb{Z} : k \geq 0\}$, where we identify $\mathrm{GF}(2)^t$ with the set of integers $\{0, 1, \ldots, 2^t - 1\}$. The kernel $V_0 = \gamma^{-1}(0)$ is the set of P-positions of the annihilation game. This gives very precise information about the structure of G: its maximum finite γ-value is a power of 2 minus 1, and the sets $\gamma^{-1}(i)$ for $i \in \{0, \ldots, 2^t - 1\}$ all have the same size, being cosets of V_0. Moreover, V_0 constitutes an *anncode* (annihilation game code). Though G has 2^n vertices, it turns out that most of the relevant information can be extracted from an induced subgraph of size $O(n^4)$, by an $O(n^6)$ algorithm, which is often much more efficient.

If Γ is cyclic, γ is generally distinct from the (classical) Sprague–Grundy function g on Γ; in fact, g may not even exist on Γ. Also, A played on a cyclic Γ has a distinct character and strategy from the non-annihilation game played on Γ.

Annihilation games were suggested by John Conway. Ferguson [1984] considered misère annihilation play, in which the player first unable to move wins, and the opponent loses. A more transparent presentation of annihilation games is to appear in the forthcoming book [Fraenkel $\geq$ 1997].

Section 2 gives a number of examples, illustrating connections between games, anncodes and lexicodes, as well as exponential and polynomial digraphs and computations associated with them. Section 3 gives a short proof that the Sprague–Grundy function g is linear on the lexigraph associated with lexicodes, leading to the same kind of homomorphism that exists for anncodes. Some natural further questions are posed at the end of Section 3, including the definition of anncodes over GF(q), for $q \geq 2$. Section 4 indicates, by means of a larger example, how a greedy algorithm applied to an anncode can reduce a computation of a code by a factor of 2,000 compared to a similarly computed lexicode. The anncode method is potentially polynomial, whereas the lexicode method is exponential. But it is too early yet to say to what extent the potential of the anncode method can be realized for producing new efficient codes.

2. Examples

Given a finite digraph $G = (V, E)$, we define, for any $u \in V$, the set of *followers* $F(u)$ and *ancestors* $F^{-1}(u)$ by

$$F(u) = \{v \in V : (u,v) \in E\}, \qquad F^{-1}(u) = \{w \in V : (w,u) \in E\}.$$

If we regard the vertices of G as game positions and the edges as moves, we define, as usual, a *P-position* of the game as one from which the Previous player can win, no matter how the opponent plays, subject to the rules of the game; an *N-position* is one that is a Next-player win. Denote by $\mathcal{P}$ the sets of all P-positions of a game, and denote by $\mathcal{N}$ the set of all N-positions. The following basic relationships hold:

$$u \in \mathcal{P} \quad \text{if and only if} \quad F(u) \subseteq \mathcal{N},$$
$$u \in \mathcal{N} \quad \text{if and only if} \quad F(u) \cap \mathcal{P} \neq \varnothing.$$

If G has cycles or loops, the game may also contain dynamically drawn *D-positions*; the set $\mathcal{D}$ of such positions is characterized by

$$u \in \mathcal{D} \quad \text{if and only if} \quad F(u) \subseteq \mathcal{D} \cup \mathcal{N} \text{ and } F(u) \cap \mathcal{D} \neq \varnothing.$$

To understand the examples below we don't need γ or g; it suffices to know that $\mathcal{P}$ is the set of vertices on which γ or g is 0. Note that $\mathcal{P}$ can be recognized by purely game-theoretic considerations, as the set on which the Previous player

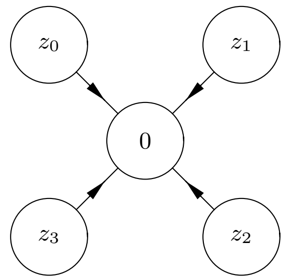

Figure 1. An acyclic groundgraph for annihilation (see Example 2.1).

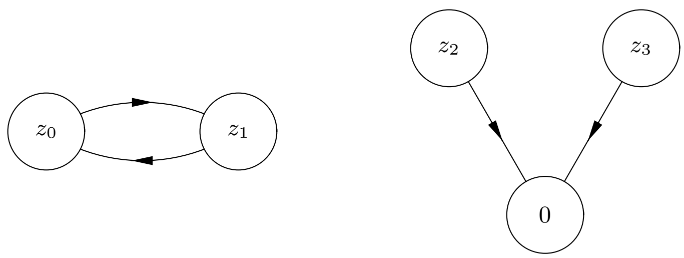

Figure 2. A cyclic groundgraph (see Example 2.2).

can win. In all these examples, we play an annihilation game A on the given groundgraphs Γ.

Example 2.1. Let Γ be the digraph depicted in Figure 1. It is easy to see that with an odd number of tokens on the z_i the first player can win, and with an even number the second player can win in A played on Γ.

In this and the following examples, think of the z_i as unit vectors of a vector space V of dimension n, where $n-1$ is the largest index of the z_i [Fraenkel and Yesha 1982]. In the present example, $z_0 = (0001), \ldots, z_3 = (1000)$. Encoded by the unit vectors, our anncode is

$$\mathcal{P} = \big\{(0000), (0011), (0101), (0110), (1001), (1010), (1100), (1111)\big\},$$

or, encoded in decimal, $\mathcal{P} = \{0, 3, 5, 6, 9, 10, 12, 15\}$. Note that $\mathcal{P}$ is a linear code with minimal Hamming distance $d = 2$.

(Recall that the *Hamming distance* between two vectors in $\mathrm{GF}(2)^n$ is the number of 1-bits of their difference. The number of 1-bits of a vector u is its *weight*, and is denoted by $w(u)$. Addition, or equivalently subtraction, over $\mathrm{GF}(2)$ is denoted by $\oplus$.)

Example 2.2. Consider A played on the two-component graph Γ of Figure 2. If z_0 and z_1 host a token each, any move causes annihilation. Therefore the

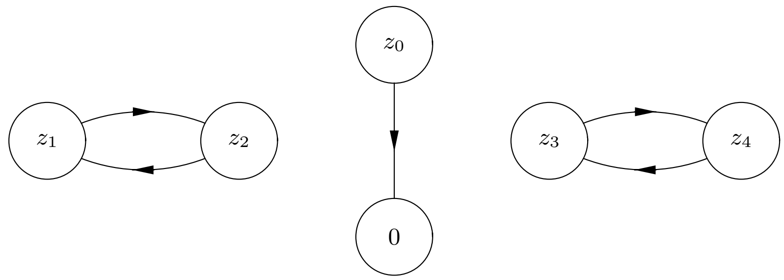

Figure 3. Another cyclic groundgraph (see Example 2.3).

position consisting of one token each on z_0, z_1, z_2 (or z_3 instead of z_2) is a P-position. Using decimal encoding, we then see that $\mathcal{P} = \{0, 7, 11, 12\}$, which is also a linear code with minimal distance 2.

EXAMPLE 2.3. Consider A played on a Nim-heap of size 5, i.e., Γ consists of the leaf 0 and the vertices $z_0, \dots, z_4$, where $(z_j, z_i) \in E(\Gamma)$ if and only if $i < j$ and $(z_i, 0) \in E(\Gamma)$ for $i \in \{0, \dots, 4\}$. It is not hard to see that then $\mathcal{P} = \{0, 7, 25, 30\}$, which is an anncode with minimal distance 3. Precisely the same code is given by the P-positions of the annihilation game A played on the ground graph Γ of Figure 3.

In order to continue with our examples, we now define lexicodes precisely. This is also needed for Section 3.

Let W be an $n \times n$ matrix over GF(2), of rank at least m, where $m \le n$ is some integer. We will count the columns of W from the right and the rows from the bottom. Suppose the rightmost m columns of W constitute a basis of V^m, the m-dimensional vector subspace of V^n over GF(2). Then there are rows $1 \le i_1 < \dots < i_m \le n$ of W such that the $m \times m$ submatrix W_m consisting of rows $i_1, \dots, i_m$ and columns $1, \dots, m$ of W has rank m.

Construct the 2^m elements of V^m in lexicographic order:

$$V^m = \{0 = A_0, \dots, A_{2^m - 1}\}.$$

Precisely, $A_k = WK$, where K is the column vector of the binary value of $k \in \{0, \dots, 2^m - 1\}$, with the bits of K in positions $i_1, \dots, i_m$, the least significant bit in i_1; and 0's in all the other $n - m$ positions. See Table 1 for an example with $m = n$.

For given $d \in \mathbb{Z}^+$, scan V^m from A_0 to $A_{2^m - 1}$ to generate a subset $V' \subseteq V^m$ using the following greedy algorithm. Put $V' \leftarrow 0$. If $A_{i_0} = 0, \dots, A_{i_j}$ have already been inserted into V', insert $A_{i_{j+1}}$ if $i_{j+1} > i_j$ is the smallest integer such that $H(A_{i_l}, A_{i_{j+1}}) \ge d$ for $l \in \{0, \dots, j\}$, where H denotes Hamming distance. The resulting V' is the *lexicode* generated by W, with minimal distance d.

We remark that in [Brualdi and Pless 1993] the term "lexicode" is reserved for the code generated when W is the identity matrix, which is the case considered

k	V^m		V'
	BIN	DEC	
0	0000	0	$*$
1	0001	1	
2	0011	3	$*$
3	0010	2	
4	0110	6	$*$
5	0111	7	
6	0101	5	$*$
7	0100	4	
8	1100	12	$*$
9	1101	13	
10	1111	15	$*$
11	1110	14	
12	1010	10	$*$
13	1011	11	
14	1001	9	$*$
15	1000	8	

Table 1. Generating a lexicode (see Example 2.4).

in [Conway and Sloane 1986]; and "greedy codes" is used for the codes derived from any W whose columns constitute a basis. Actually, in both of these papers no matrices are used, but the ordering is done in an equivalent manner. It seems natural, in the current context, to use matrices (see the proofs in the next section) and "lexicode" for the entire class of codes.

Example 2.4. Let

$$W = \begin{array}{c} \begin{array}{cccc} 4 & 3 & 2 & 1 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \end{array},$$

and $d = 2$, $m = n = 4$. We then get the ordered vector space depicted in Table 1. The vectors marked with an asterisk in column V' have been selected by our greedy algorithm, and constitute the lexicode. Note that this lexicode is precisely the same code as that found in Example 2.1 by using a small groundgraph with $O(n^2)$ operations rather than $O(2^n)$ for the lexicode.

Example 2.5. Let

$$W = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix},$$

and $d = 2$. The reader should verify that the lexicode generated by W is $(0, 7, 12, 11)$, in this order, which is identical to the code generated in Example 2.2.

EXAMPLE 2.6. Let

$$W = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

and $d = 3$. The vector space now contains 32 entries, too large to list here. But the reader can verify that the lexicode generated by W is precisely the same as that generated by the two polynomial methods of Example 2.3.

3. The Truth About Lexicodes

We now study the Sprague–Grundy function of a certain game associated with a lexicode. Given a finite acyclic digraph $G = (V, E)$, the associated *Sprague–Grundy* function $g : V \to \mathbb{Z}^0$ is characterized by the property

$$g(u) = \operatorname{mex} g\big(F(u)\big), \tag{3.1}$$

where, for any finite subset $S \subseteq \mathbb{Z}^0$, we define $\operatorname{mex}(S) = \min(\mathbb{Z}^0 - S)$ and $g(S) = \big\{g(s) : s \in S\big\}$. This function exists uniquely on any finite acyclic digraph. See, for example, [Berge 1985, Ch. 14; 1989, Ch. 4; Conway 1976; Berlekamp, Conway and Guy 1982]. (When G has cycles or loops, g may not exist; a generalization of it, the γ-function mentioned in the introduction, can be used in this case.)

With a lexicode in V^m, with minimal distance d, associate a digraph $G = (V, E)$, called a *lexigraph*, as follows. The vertex set V is the set of all elements (vectors) of V^m, and $(A_k, A_j) \in E$ if and only if $j < k$ and

$$H(A_j, A_k) = w(A_j \oplus A_k) < d,$$

where, as before, H is the Hamming distance and w the weight. If $(A_k, A_j) \in E$, we have $A_j \in F(A_k)$ in the notation introduced at the beginning of Section 2. Note that G is finite and acyclic. (For other possibilities of orienting the lexigraph, see the homework problem towards the end of this section.)

Play a *lexigame* on G by placing a single token on any vertex. A move consists of sliding the token from a vertex to a neighboring vertex along a directed edge. The player first unable to play loses and the opponent wins. Note that any game with a single token on a digraph, and in particular the lexigame just introduced, can be considered an anngame. The P-positions of the lexigame constitute the lexicode; this is also the set of vertices of G on which $g = 0$. (Actually, the lexigame is not overly interesting, because the lexigraph is "analogous" to the game graph of a (more interesting) game played on a logarithmically smaller groundgraph with several tokens. The game graph of a game is not normally constructed, but used instead for reasoning about the game. In fact, we do this in the proof of Theorem 3.9 below.)

We point out that for lexicodes *per se* it suffices to consider the case $m = n$. It is only in Corollary 3.7 and in Section 4, where we apply a greedy algorithm on anncodes, that the case $m < n$ will be important. Incidentally, Brualdi and Pless [1993, § 2] define a function g and state, citing [Conway and Sloane 1986], that g is the Sprague–Grundy function of an associated heap game for the case where W is the unit matrix. It is easy to see that, in fact, g is the Sprague–Grundy function of the lexigraph defined above, for every matrix W.

For any positive integer s, let s^h denote the bit in the h-th binary position of the binary expansion of s, where s^0 denotes the least significant bit. Also, for any $a \in \mathbb{Z}^0$, write $\phi(a) = \{0, \ldots, a-1\}$.

LEMMA 3.1. *Let $a_1, a_2 \in \mathbb{Z}^0$, and let $b \in \phi(a_1 \oplus a_2)$. Then there is $i \in \{1, 2\}$ and $d \in \phi(a_i)$ such that $b = a_j \oplus d$ for $j \neq i$.*

PROOF. Write $c = a_1 \oplus a_2$. Let $k = \max\{h : b^h \neq c^h\}$. Since $b < c$, we have $b^k = 0$ and $c^k = 1$. Hence there exists $i \in \{1, 2\}$ such that $a_i^k = 1$. Letting $d = a_i \oplus b \oplus c = a_j \oplus b$, we have $d \in \phi(a_i)$, since $b^h = c^h$ implies $d^h = a_i^h$ for $h > k$, and $d^k = 0$. □

COROLLARY 3.2. *We have $\phi(a_1 \oplus a_2) \subset a_1 \oplus \phi(a_2) \cup \phi(a_1) \oplus a_2$.* □

By the closure of V^m, for any j and k there exists l such that $A_j \oplus A_k = A_l$.

LEMMA 3.3. *We have $A_j \oplus A_k = A_{j \oplus k}$.*

PROOF. As noted above, $A_j \oplus A_k = A_l$ for some l. Then $A_j = WJ$, $A_k = WK$, $A_l = WL$. Thus

$$WL = A_l = A_j \oplus A_k = W(J \oplus K).$$

This matrix equation implies $W_m L_m = W_m(J_m \oplus K_m)$, where W_m was defined in Section 2, and any $m \times 1$ vector X_m is obtained from the $n \times 1$ vector X by retaining only the rows $i_1, \ldots, i_m$ of X and deleting the $n - m$ remaining rows, which contain only 0's for L, J and K. Since W_m is invertible, we thus get $L_m = J_m \oplus K_m$, so $l = j \oplus k$. □

Here is the main lemma of this section.

LEMMA 3.4. *Let $A_j, A_k \in V^m$. Then, for the lexigraph on V^m,*

$$F(A_j \oplus A_k) \subseteq A_j \oplus F(A_k) \cup F(A_j) \oplus A_k \subseteq F(A_j \oplus A_k) \cup F^{-1}(A_j \oplus A_k).$$

PROOF. Let $A_l \in F(A_j \oplus A_k)$. By Lemma 3.3, $A_l \in F(A_{j \oplus k})$, so $w(A_l \oplus A_{j \oplus k}) = w(A_{j \oplus k \oplus l}) < d$ and $l < j \oplus k$. By Corollary 3.2, $l \in j \oplus \phi(k) \cup \phi(j) \oplus k$. Thus either there is $k' < k$ such that $l = j \oplus k'$, or there is $j' < j$ such that $l = j' \oplus k$. In the former case, $w(A_{j \oplus k \oplus l}) = w(A_{k \oplus k'}) < d$, so $A_l = A_j \oplus A_{k'} \in A_j \oplus F(A_k)$, and in the latter case we obtain, similarly, $A_l \in F(A_j) \oplus A_k$, establishing the left inclusion.

Now let $A_l \in A_j \oplus F(A_k) \cup F(A_j) \oplus A_k$. Then either $A_l = A_j \oplus A_{k'}$ for some $k' < k$ with $w(A_{k \oplus k'}) < d$, or $A_l = A_{j'} \oplus A_k$ for some $j' < j$ with $w(A_{j \oplus j'}) < d$. Without loss of generality, assume the former. Then $l = j \oplus k'$. Thus $w(A_{k \oplus k'}) = w(A_{j \oplus k \oplus l}) < d$. If $l < j \oplus k$, then $A_l \in F(A_j \oplus A_k)$, and if $l > j \oplus k$, then $A_j \oplus A_k \in F(A_l)$. □

We now show that the g-function is linear on the lexigraph G.

Theorem 3.5. *Let $G = (V, E)$ be a lexigraph. Then $g(u_1 \oplus u_2) = g(u_1) \oplus g(u_2)$ for all $u_1, u_2 \in V$.*

Proof. Set

$$\mathcal{F}(u_1, u_2) = \{u_1\} \times F(u_2) \cup F(u_1) \times \{u_2\}, \tag{3.2}$$

so that $(v_1, v_2) \in \mathcal{F}(u_1, u_2)$ if either $v_1 = u_1$ and $v_2 \in F(u_2)$, or $v_1 \in F(u_1)$ and $v_2 = u_2$: Thus $\mathcal{F}$ represents the set of followers in the sum game played on $G + G$. Let

$$K = \{(u_1, u_2) \in V \times V : g(u_1 \oplus u_2) \neq g(u_1) \oplus g(u_2)\},$$
$$k = \min_{(u_1, u_2) \in K} \big(g(u_1 \oplus u_2),\ g(u_1) \oplus g(u_2)\big).$$

If there is $(u_1, u_2) \in K$ such that $g(u_1 \oplus u_2) = k$, then $g(u_1) \oplus g(u_2) > k$. By Corollary 3.2 and the mex property (3.1) of g, there is $(v_1, v_2) \in \mathcal{F}(u_1, u_2)$ such that $g(v_1) \oplus g(v_2) = k$. Now (3.2) implies

$$v_1 \oplus v_2 \in u_1 \oplus F(u_2) \cup F(u_1) \oplus u_2 \subseteq F(u_1 \oplus u_2) \cup F^{-1}(u_1 \oplus u_2),$$

where the inclusion follows from Lemma 3.4. Since $g(u_1 \oplus u_2) = k$, it follows that $g(v_1 \oplus v_2) > k$, so $(v_1, v_2) \in K$. Let

$$L = \{(u_1, u_2) \in K : g(u_1) \oplus g(u_2) = k\}.$$

We have just shown that $K \neq \varnothing$ implies $L \neq \varnothing$.

Here we recall that g is the γ-function for the lexigame (see the first paragraph of this section). With a γ-function we can associate a monotonic counter function $c : V \to \mathbb{Z}^+$. We now pick $(u_1, u_2) \in L$ with $c(u_1) + c(u_2)$ minimal. For $(u_1, u_2) \in L$ we have $g(u_1 \oplus u_2) > k$. Then there is $v \in F(u_1 \oplus u_2)$ with $g(v) = k$. By the first inclusion of Lemma 3.4, there exists $(v_1, v_2) \in \mathcal{F}(u_1, u_2)$ such that $v = v_1 \oplus v_2$. So $g(v_1 \oplus v_2) = k$. Since $g(u_1) \oplus g(u_2) = k$, (3.2) implies $g(v_1) \oplus g(v_2) > k$, hence $(v_1, v_2) \in K$. As we saw earlier, this implies that there is $(w_1, w_2) \in \mathcal{F}(v_1, v_2)$ such that $(w_1, w_2) \in L$. Moreover, by property B in the definition of the γ-function (see [Fraenkel 1996, p. 20] in this volume), we can select (w_1, w_2) such that $c(w_1) + c(w_2) < c(u_1) + c(u_2)$, contradicing the minimality of $c(u_1) + c(u_2)$. Thus $L = K = \varnothing$. □

Let $V_i = \{u \in V : g(u) = i\}$, for $i \geq 0$. We now state the main result of this section.

Theorem 3.6. *Let $G = (V, E)$ be a lexigraph. Then $V_0 = V'$, where V' is a lexicode. Moreover, V_0 is a linear subspace of V. In fact, g is a homomorphism from V onto* $\mathrm{GF}(2)^t$ *for some $t \in \mathbb{Z}^0$; its kernel is V_0, and the quotient space V/V_0 consists of the cosets V_i for $0 \leq i < 2^t\}$; in fact, $t = \dim V - \dim V_0$.*

Proof. By definition, V is a vector space over GF(2). Let t be the smallest nonnegative integer such that $g(u) \leq 2^t - 1$ for all $u \in V$. Thus, if $t \geq 1$, there is some $v \in V$ such that $g(v) \geq 2^{t-1}$. Then the "1's complement" of $g(v)$, defined as $2^t - 1 - g(v)$, is less than $g(v)$. By the mex property of g, there exists $w \in F(v)$ such that $g(w) = 2^t - 1 - g(v)$. By Theorem 3.5, $g(v \oplus w) = g(v) \oplus g(w) = 2^t - 1$. Thus, again by the mex property of g, every value in $\{0, \ldots, 2^t - 1\}$ is the g-value of some $u \in V$. This last property holds trivially also for $t = 0$. Hence g is onto. It is a homomorphism $V \to \mathrm{GF}(2)^t$ by Theorem 3.5, and since $g(1u) = g(u) = 1g(u)$ and $g(0u) = g(0 \ldots 0) = 0 = 0g(u)$.

By elementary linear algebra, $\mathrm{GF}(2)^t \simeq V/V_0$, where V_0 is the kernel. Hence V_0 is a subspace of V. Clearly V_0 is also a graph-kernel of G. So is V', which, by its definition, is both independent and dominating. Since any finite acyclic digraph has a unique kernel, $V_0 = V'$. Let $m = \dim V_0$. Then $\dim V = m + t$. The elements of V/V_0 are the cosets $V_i = w \oplus V_0$ for any $w \in V_i$ and every $i \in \{0, \ldots, 2^t - 1\}$. □

Corollary 3.7. *The greedy algorithm, applied to any lexicographic ordering of the subset $V_0 \subset V$, also produces a linear code.*

Proof. Follows from Theorem 3.6, by considering the lexigraph $G = (V_0, E)$ instead of (V, E). □

We remark that Algorithm B of [Fraenkel and Yesha 1982] yields a matrix Γ, whose bottom $n - m$ rows, padded with m bottom 0-rows, is the parity check matrix for the code (vectors where $\gamma = 0$). A much simplified version of this algorithm can be used to compute the parity check matrix for the present case (vectors where $g = 0$).

Homework 3.8. The lexigraph $G = (V, E)$ seems to exhibit a certain robustness, roughly speaking, with respect to E. That is, Theorem 3.6 seems to be invariant under certain edge deletions or reversions. In this direction, prove that Theorem 3.6 is still valid if E is defined as follows: $(A_k, A_j) \in E$ if and only if $A_j < A_k$ (rather than $j < k$) and $H(A_j, A_k) < d$.

Theorem 3.9. *The set of lexicodes is a subset of the set of anncodes.*

Proof. Let C be a lexicode with a given minimal distance. As we saw at the beginning of this section, C is the set of the P-positions of the lexigame played on the lexigraph G, or equivalently the set of vertices where the Sprague–Grundy function g is zero. The lexigame is played on G by sliding a single token, and as such it is an annihilation game; the anncode is the set of vertices where the generalized Sprague–Grundy function γ is zero. The two functions are the same, since the graph is acyclic. Thus C is an anncode. □

The proof of Theorem 3.6 is actually a much simplified version of a similar result for annihilation games [Fraenkel and Yesha 1982], where also the linearity of γ (and hence of g) was proved for the first time, to the best of our knowledge. The simplification in the proof is no accident, since the lexigame played on the lexigraph (the groundgraph) can be considered as an anngame with a single token. It's an acyclic groundgraph, which makes the anngame theory much simpler than for cyclic digraphs.

It might be of interest to explore the subset of anncodes generated when several tokens, rather than only one, are distributed initially on a lexigraph.

Another question is: Under what conditions, and for what finite fields $\mathrm{GF}(p^a)$, where p is prime and $a \in \mathbb{Z}^+$, are there "anncodes"? The key seems to be to generalize annihilation games as follows. On a given finite digraph Γ, place nonzero "particles" (elements of $\mathrm{GF}(p^a)$), at most one particle per vertex. A move consists in selecting an occupied vertex and moving its particle to a neighboring vertex v along a directed edge. If v was occupied, then the "collision" generates a new particle, possibly 0 ("annihilation"), according to the addition table of $\mathrm{GF}(p^a)$. The special case $a = 1$, when the particles are $0, \ldots, p-1$, reduces to *p-annihilation*: the collision of particles i and j results in particle k, where $k \equiv i + j \bmod p$, for $k < p$; and this special case becomes anngames for $p = 2$. Such "Elementary Particle Physics" games, whose P-positions are *collections* of linear codes, thus constitute a generalization of anngames. These games and their applications to coding seems to be an as yet unexplored area.

4. Computing Anncodes

In this section we give one particular example illustrating the computation of large anncodes. One can easily produce many others. The present example also shows how anncodes and lexicodes can be made to join forces.

We begin with a family Γ_t of groundgraphs, which is a slightly simplified version of a family considered by Yesha [1978] for showing that the finite γ-values on an annihilation game played on a digraph without leaves can be arbitrarily large.

Let $t \in \mathbb{Z}^+$, and set $J = J(t) = 2^{t-1}$. The digraph Γ_t has vertex set $\{x_1, \ldots, x_J, y_1, \ldots, y_J\}$, and edges as follows:

$$F(x_i) = y_i \qquad \text{for } i = 1, \ldots, J,$$
$$F(y_k) = \{y_i : 1 \le i < k\} \cup \{x_j : 1 \le j \le J \text{ and } j \ne k\} \qquad \text{for } k = 1, \ldots, J.$$

Figure 4 shows Γ_3.

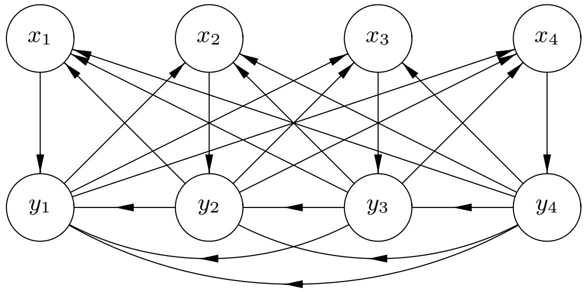

Figure 4. The cyclic groundgraph Γ_3.

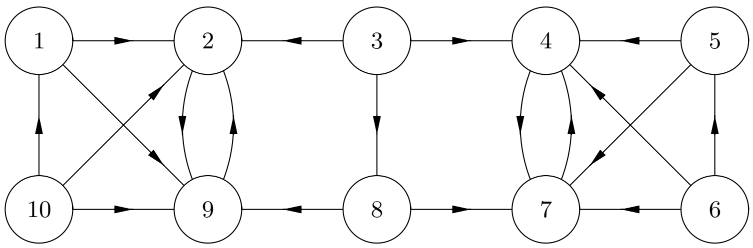

Figure 5. The cyclic groundgraph Γ'.

Since Γ_t has no leaf, $\gamma(x_i) = \gamma(y_i) = \infty$ for all $i \in \{1, \ldots, J\}$. The following facts about the anngraph $G_t = (V, E)$ of Γ_t are easy to establish, where $V^f = \{\boldsymbol{u} \in V : \gamma(\boldsymbol{u}) < \infty\}$.

(i) $\gamma(x_i \oplus x_j) = 0$ for all $i \neq j$.
(ii) $\gamma(x_i \oplus y_j) = j$ for all $i, j \in \{1, \ldots, J\}$.
(iii) $\gamma(y_i \oplus y_j) = i \oplus j$ for all $i \neq j$.
(iv) $\max_{\gamma(\boldsymbol{u})<\infty} \gamma(\boldsymbol{u}) = \gamma(y_{J-1} \oplus y_J) = (J-1) \oplus J = 2^t - 1$.
(v) $V^f = \{\boldsymbol{u} \in V : w(\boldsymbol{u}) \equiv 0 \bmod 2\}, \quad |V^f| = 2^{2J-1}, \quad \dim V^f = 2J - 1$.

Thus, in the notation of Theorem 3.6, we have $m + t = 2J - 1$, hence

$$m = 2^t - t - 1.$$

For the family Γ_t of groundgraphs, the $O(n^6)$ algorithm for computing γ thus reduces to an $O(1)$ algorithm.

Now consider the groundgraph Γ' depicted in Figure 5. It is not hard to see that a basis for V_0 is given by the four vectors $(1, 2, 9, 10)$, $(4, 5, 6, 7)$, $(2, 3, 8, 9)$, $(3, 4, 7, 8)$. Each vector indicates the four vertices occupied by tokens.

We propose to play an annihilation game, say on $\Gamma = \Gamma_5 + \Gamma'$, which contains $32 + 10 = 42$ vertices. The vector space associated with the anngraph of Γ contains 2^{42} elements, and to find a lexicode on V^{42}, for any given d, involves 2^{42} operations. On the other hand, for Γ we have, since $t = 1$ for Γ',

$$\dim |V_0| = m = 2^5 - 5 - 1 + 4 + 1 = 31,$$

so the anncode defined by V_0, for which $d = 2$, has 2^{31} elements. By the results of Section 2, we can compute a *lexi-anncode* for any $d > 2$, by applying the greedy algorithm to a lexicographic ordering of V_0, which can be obtained by using any basis of V_0. This computation involves only 2^{31} operations.

HOMEWORK 4.1. Carry out this computation, and find lexi-anncodes for several $d > 2$ on $\Gamma = \Gamma_5 + \Gamma'$.

We note that the anncode derived from a directed complete graph, i.e., a Nim-heap, is identical to the code derived from certain coin-turning games as considered in [Berlekamp, Conway and Guy 1982, Ch. 14].

REMARK 4.2. The Hamming distance between any two consecutive P-positions in an annihilation game is obviously ≤ 4. Thus $d = 2$ for Γ_t and $d = 4$ for Γ'. For finding codes with $d > 4$, it is thus natural to apply the greedy algorithm to a lexicographic ordering of V_0. Another method to produce anncodes with $d > 4$ is to encode each vertex of the groundgraph, that is, each bit of the anngraph, by means of k bits for some fixed $k \in \mathbb{Z}^+$. For example, in a lexigraph, each vertex is encoded by n bits, and the distance between any two codewords is $\geq d$. In an Elementary Particle Physics game over $\mathrm{GF}(p^a)$, it seems natural to encode each *particle* by a digits. A third method for producing anncodes with $d > 4$ directly seems to be to consider a generalization of anngames to the case where a move consists of sliding precisely k (or $\leq k$) tokens, where k is a fixed positive integer parameter—somewhat analogously to Moore's Nim (see [Berlekamp, Conway and Guy 1982, Ch. 15], for example).

REMARK 4.3. Note that $\bigcup_{i=0}^{2^k-1} V_i$ is a linear subspace of V^f for every $k \in \{0, \ldots, t\}$. Any of these subspaces is thus also a linear code, in addition to V_0.

Acknowledgment

This paper is a direct result of Vera Pless's lecture at the Workshop on Combinatorial Games held at MSRI, Berkeley, CA, in July, 1994. I had intended to write it after I heard her lecture at the AMS Short Course on Combinatorial Games held in Columbus, OH, in the summer of 1990. But I put it off. This time, at H^sLordship's banquet in Berkeley, Herb Wilf challenged me with his insights and comments, rekindling my interest in this topic. I also had a shorter conversation about it with John Conway and Bill Thurston at that MSRI meeting. Actually, I had originally discussed the possibility of using codes derived from annihilation games with C. L. Liu and E. M. Reingold in 1978 or 1979, and again briefly with Richard Guy in the eighties and with Ya'acov Yesha in the early nineties.

References

1. [Berge 1985] C. Berge, *Graphs*, North-Holland, Amsterdam, 1985.
2. [Berge 1989] C. Berge, *Hypergraphs: Combinatorics of Finite Sets*, North-Holland and Elsevier, Amsterdam, 1989. Original French edition: *Hypergraphes: combinatoire des ensembles finis*, Gauthier-Villars, Paris, 1987.
3. [Berlekamp, Conway and Guy 1982] E. R. Berlekamp, J. H. Conway and R. K. Guy, *Winning Ways for Your Mathematical Plays*, Academic Press, London, 1982.
4. [Brualdi and Pless 1993] R. A. Brualdi and V. S. Pless, "Greedy codes", *J. Combin. Theory* A**64** (1993), 10–30.
5. [Conway 1976] J. H. Conway, *On Numbers and Games*, Academic Press, London, 1976.
6. [Conway 1990] J. H. Conway, "Integral lexicographic codes", *Discrete Math.* **83** (1990), 219–235.
7. [Conway and Sloane 1986] J. H. Conway and N. J. A. Sloane, "Lexicographic codes: error-correcting codes from game theory", *IEEE Trans. Inform. Theory* IT-**32** (1986), 337–348.
8. [Ferguson 1984] T. S. Ferguson, "Misère annihilation games", *J. Combin. Theory* A**37** (1984), 205–230.
9. [Fraenkel 1974] A. S. Fraenkel, "Combinatorial games with an annihiliation rule", pp. 87–91 in *The Influence of Computing on Mathematical Research and Education* (edited by J. P. LaSalle), Proc. Symp. Appl. Math. **20**, Amer. Math. Soc., Providence, RI, 1974.
10. [Fraenkel 1996] A. S. Fraenkel, "Scenic trails ascending from sea-level Nim to alpine chess", pp. 13–42 in this volume.
11. [Fraenkel $\geq$1997] A. S. Fraenkel, *Adventures in Games and Computational Complexity*, to appear in Graduate Studies in Mathematics, Amer. Math. Soc., Providence, RI.
12. [Fraenkel and Perl 1975] A. S. Fraenkel and Y. Perl (1975), "Constructions in combinatorial games with cycles", pp. 667–699 in *Proc. Intern. Colloq. on Infinite and Finite Sets* (edited by A. Hajnal et al.), vol. 2, Colloq. Math. Soc. János Bolyai **10** North-Holland, Amsterdam, 1975.
13. [Fraenkel, Tassa and Yesha 1978] A. S. Fraenkel, U. Tassa and Y. Yesha, "Three annihilation games", *Math. Mag.* **51** (1978), 13–17.
14. [Fraenkel and Yesha 1976] A. S. Fraenkel and Y. Yesha, "Theory of annihilation games", *Bull. Amer. Math. Soc.* **82** (1976), 775–777.
15. [Fraenkel and Yesha 1979] A. S. Fraenkel and Y. Yesha, "Complexity of problems in games, graphs and algebraic equations", *Discrete Appl. Math.* **1** (1979), 15–30.
16. [Fraenkel and Yesha 1982] A. S. Fraenkel and Y. Yesha, "Theory of annihilation games – I", *J. Combin. Theory* B**33** (1982), 60–86.
17. [Fraenkel and Yesha 1986] A. S. Fraenkel and Y. Yesha, "The generalized Sprague–Grundy function and its invariance under certain mappings", *J. Combin. Theory* A**43** (1986), 165–177.

18. [Smith 1966] C. A. B. Smith, "Graphs and composite games", *J. Combin. Theory* **1** (1966), 51-81. Reprinted in slightly modified form in *A Seminar on Graph Theory* (edited by F. Harary), Holt, Rinehart and Winston, New York, 1967.

19. [Yesha 1978] Y. Yesha, "Theory of annihilation games", Ph.D. Thesis, Weizmann Institute of Science, Rehovot (Israel), 1978.

Aviezri S. Fraenkel
Department of Applied Mathematics and Computer Science
Weizmann Institute of Science
Rehovot 76100, Israel
fraenkel@wisdom.weizmann.ac.il
http://www.wisdom.weizmann.ac.il/~fraenkel/fraenkel.html