Perfect One Error Correcting Codes on Iterated Complete Graphs

Be Birchall
Reed College
Portland, Oregon
ebirchal@reed.edu

jason Tedor
University of Alaska Fairbanks
Fairbanks, Alaska
fsjet1@uaf.edu

August 24, 1999

Abstract

Given an arbitrary graph, a perfect one error correcting code is a subset of the vertices called codewords such that no two codewords are adjacent and every non-codeword is adjacent to exactly one codeword. Determining if there is a perfect one error correcting code on an arbitrary graph seems difficult; in fact, it is NP-Complete. We present a biinfinite family of graphs based on the complete graphs such that there is a unique perfect one error correcting code on every graph in the family. We present recursive constructions of these graphs and constructions for determining which vertices are codewords. Given an arbitrary finite alphabet, we show how to assign the strings of fixed length over that alphabet to a graph in the family. This assignment is such that determining which strings correspond to codewords is easy. 'Easy' here means that codeword recognition can be accomplished by a four state finite state machine. Moreover, error-correction can be accomplished by a finite state machine. The code which we present is nonlinear yet codeword recognition and error-correction can be accomplished as easily as linear codes.

Contents

1 Introduction

The idea of a perfect one error correcting code is a generalization of the error correcting codes on the hypercube. In this paper we explore the idea of a perfect one error correcting code. This exploration will be tied to a biinfinite family of graphs.

2 Definitions

2.1 Perfect One Error Correcting Codes

Definition 2.1. A *coded graph* is an ordered pair $H = (G, C)$ where $G = (V, E)$ is a graph and $C \subset V$.

The elements of C are called codewords; the elements of $V \setminus C$ are called non-codewords. We say C is a code on G and refer to H as a coded G.

To simplify our discussion, we often refer to graph theoretic properties of H by referring directly to H rather than referring to the underlying graph G. For example, instead of saying K is a subgraph of G we say K is a subgraph of H. The meaning of such statements will always be clear from context.

Definition 2.2. Given coded graphs $H = (G_1, C_1)$ and $K = (G_2, C_2)$ where $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, H is *codeword isomorphic* to K if there is a $\varphi : V_1 \to V_2$ such that

1. φ is a graph isomorphism from G_1 to G_2
2. $c \in C_1$ if and only if $\varphi(c) \in C_2$.

Definition 2.3. Given a coded graph (G, C), C is a *perfect one error correcting code* on G if

1. no two codewords are adjacent
2. every non-codeword is adjacent to exactly one codeword.

Determining if there is a perfect one error correcting code on G is in general a difficult problem. We state this formally.

Definition 2.4. The problem of deciding whether or not there is a perfect one error correcting code on a given graph G is the *P1ECC decision problem.*

Cull and Nelson [CN99] show the P1ECC decision problem is NP-Complete by transforming 3-SAT to P1ECC.

2.2 Iterated Complete Graphs

Definition 2.5. The graph with n vertices such that every vertex is adjacent to every other vertex is the *complete graph on n vertices* and is denoted K_n.

The iterated complete graphs are a biinfinite family of graphs based on the complete graph on n vertices. Let Z_n^1 be the complete graph on n vertices. To construct Z_n^m for $m > 1$ first form n copies of Z_n^{m-1}. Then, choose $n-1$ vertices of minimal degree from each of the n copies of Z_n^{m-1}. Now form $\binom{n}{2}$ edges $\{x, y\}$ where x and y are from our chosen $n^2 - n$ vertices such that

1. there is exactly one edge between any two distinct copies of Z_n^{m-1}

2. if $\{x, y\}$ and $\{x, z\}$ are edges then $y = z$.

Item two in the above implies that for every $v \in V(Z_n^m), \deg v < n + 1$.

Informally Z_n^2 can be thought of as a complete graph of complete graphs. That is, Z_n^2 can be thought of as a K_n where the "vertices" are copies of K_n. Similarly, Z_n^3 can be thought of as a K_n where the "vertices" are copies of Z_n^2. In general, Z_n^m can be thought of as a K_n where the "vertices" are Z_n^{m-1}. The figures on the following page should clarify this idea and the construction.

Definition 2.6. A vertex $v \in V(Z_n^m)$ is *corner vertex* if $\deg v = n - 1$.

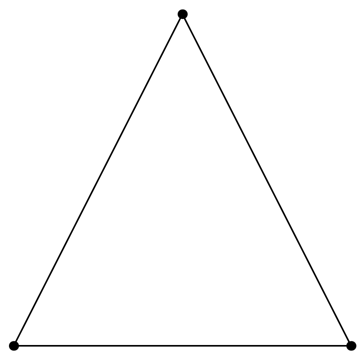

Figure 1: Z_3^1

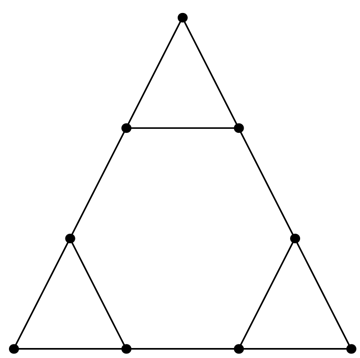

Figure 2: Z_3^2

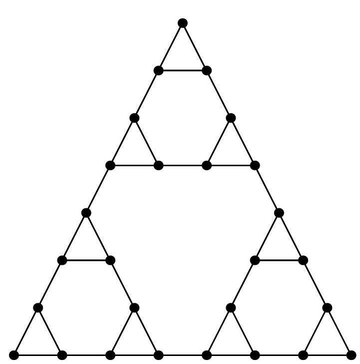

Figure 3: Z_3^3

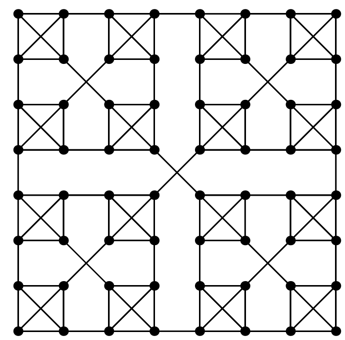

Figure 4: Z_4^3

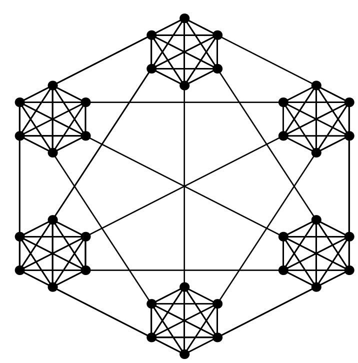

Figure 5: Z_6^2

Figure 6: Z_2^3

3 Existence and Uniqueness

3.1 An Alternate Construction of Z_n^m

We now describe a new method of constructing the iterated complete graphs.
Let ζ_n^1 be the complete graph on n vertices. To construct ζ_n^m for $m > 1$ first form a copy of K_n for each $v \in \zeta_n^m$. Denote the copy associated with vertex v $K_n(v)$. Form edges between the copies of K_n so that

1. there is an edge incident on a vertex of both $K_n(u)$ and $K_n(v)$ if and only if $\{u, v\} \in E(\zeta_n^m)$
2. the degree of every vertex is less than $n + 1$.

Theorem 3.1. For every m and n, $Z_n^m = \zeta_n^m$.

Proof. Clearly $Z_n^m = \zeta_n^m$ for $m = 1, 2$. Suppose $Z_n^k = \zeta_n^k$ for all k such that $1 \leq k \leq m$ for some m. We show this implies $Z_n^{m+1} = \zeta_n^{m+1}$. Since $Z_n^m = \zeta_n^m, \zeta_n^m$ consists of n copies of $Z_n^{m-1} = \zeta_n^{m-1}$. Since performing the construction method on ζ_n^{m-1} yields $\zeta_n^m = Z_n^m$, performing the construction method on n copies of ζ_n^{m-1} will yield n copies of $\zeta_n^m = Z_n^m$. It remains only to show that there is exactly one edge between distinct copies of Z_n^m and the degree of every vertex is less than $n + 1$. Both are clear. $\square$

3.2 Existence

Definition 3.2. Given a coded graph $H = (G, C)$, a subgraph K of G is *blank* if for every $v \in V(K), v \notin C$.

Definition 3.3. Given a graph $G = (V, E)$ and subgraphs H and K, we say H is *adjacent* to K if there is $x \in V(H)$ and $y \in V(K)$ such that $\{x, y\} \in E$. In this case we say x *joins* H to K.

We now define two families of coded Z_n^m, G_n^m and U_n^m. Let $C(G_n^1)$ be such that $|C(G_n^1)| = 1$ and let $C(U_n^1) = \emptyset$. To construct $C(U_n^{m+1})$ given $C(G_n^m)$ let $x \in C(U_n^{m+1})$ if and only if there is a $u \in V(Z_n^m)$ such that $x \in K_n(u)$ and

1. there is a $v \in C(G_n^m)$ such that x joins $K_n(u)$ to $K_n(v)$
   or
2. $\deg x = n - 1$ and there is no $v \in C(G_n^m)$ such that $K_n(u)$ is adjacent to $K_n(v)$.

To construct $C(G_n^{m+1})$ given $C(U_n^m)$ replace '$C(U_n^{m+1}$' with '$C(G_n^{m+1})$' and '$C(G_n^m)$' with '$C(U_n^m)$' in the above description. We state this for completeness.
To construct $C(G_n^{m+1})$ given $C(U_n^m)$ let $x \in C(G_n^{m+1})$ if and only if there is a $u \in V(Z_n^m)$ such that $x \in K_n(u)$ and

1. there is a $v \in C(U_n^m)$ such that x joins $K_n(u)$ to $K_n(v)$
   or

2. $\deg x = n-1$ and there is no $v \in C(U_n^m)$ such that $K_n(u)$ is adjacent to $K_n(v)$.

We state a few facts which may give the gentle reader a more intuitive feel for the above construction.

1. A subgraph $K_n(v)$ is blank in G_n^{m+1} if and only if $v \in C(U_n^m)$.
2. A vertex $x \in C(G_n^{m+1})$ if and only if x joins a non-blank $K_n(u)$ to a blank $K_n(v)$ or x is a corner vertex and x is in a non-blank $K_n(u)$ not adjacent to any blank $K_n(v)$.

Swapping the roles of 'G' and 'U' in the above yields similar results concerning the construction of U_n^m.

Lemma 3.4. The following hold for G_n^m and U_n^m for every m and n:

1. no two codewords are adjacent
2. every non-codeword is adjacent to at most one codeword.

Proof. Suppose the properties hold for U_n^{m-1}. We use this to show they hold for G_n^m.

1. Suppose for a moment there are adjacent codewords x and y in G_n^m. Let u and v be such that $x \in K_n(u)$ and $y \in K_n(v)$.

   Case 1 ($u = v$): Suppose x and y are non-corner vertices. Then there are $s, t \in V(U_n^{m-1})$ such that x joins $K_n(u)$ to blank $K_n(s)$ and y joins $K_n(u)$ to blank $K_n(t)$. Note $s \neq t$ since there are never two edges between two distinct copies of K_n. Hence, u is adjacent to two codewords s and t contradicting the induction hypothesis.

   If it is not the case that x and y are both not corner vertices, we may assume without loss of generality that x is a corner vertex and y is non-corner vertex. Hence y joins $K_n(u)$ to a blank K_n. But then $x \notin C(G_n^m)$, a contradiction.

   Case 2 ($u \neq v$): Since $u \neq v$, x joins $K_n(u)$ to $K_n(v)$. Hence, $\deg x = n$ so $K_n(v)$ is blank, a contradiction.

2. Suppose there is an $x \in V(G_n^m)$ such that x is adjacent to distinct codewords y and z. Let $u, v, w \in V(U_n^{m-1})$ be such that $x \in K_n(u), y \in K_n(v)$ and $z \in K_n(w)$. Suppose $u \neq v$ and $u \neq w$. Then $\deg x = n-1+2 = n+1$, a contradiction. Hence $u = v$ or $u = w$. Without loss of generality, say $u = w$. If $u = v$ then y is adjacent z which contradicts the above that no two codewords are adjacent. So $u \neq v$. Since $\deg y = n$, $K_n(u)$ must be blank, a contradiction.

A similar argument will establish that if the above properties hold for G_n^{m-1} then they will hold for U_n^m. $\square$

Lemma 3.5. Every non-codeword in G_n^m is adjacent to at least one codeword.

Proof. We can easily verify the claim holds for $m = 1$ and $m = 2$. Suppose the claim holds for G_n^{m-2}. Let x be a non-codeword in $V(G_n^m)$. Let $u \in U^{m-1}$ be such that $x \in K_n(u)$.

Case 1 (x is not a corner vertex): If x is not a corner vertex let $y \in V(G_n^m)$ and $v \in U_n^{m-1}$ be such that y joins $K_n(v)$ to $K_n(u)$. If x is not adjacent to a codeword in $K_n(u)$, then $K_n(u)$ is blank. Hence, y is a codeword.

Case 2 (x is a corner vertex): Let $v \in V(G_n^{m-2})$ be such that $u \in K_n(v)$.

If v is a codeword then $K_n(v)$ is blank. Hence, $K_n(u)$ is a nonblank and for every w such that $K_n(u)$ is adjacent to $K_n(w), K_n(w)$ is blank. Hence, x is a codeword.

If v is not a codeword then v is adjacent to a codeword y. Then $K_n(y)$ is blank so there is a $z \in K_n(v)$ such that z joins $K_n(v)$ to $K_n(y)$ and z is a codeword. Then, $K_n(z)$ is blank so there is a $t \in K_n(v)$ such that t is a codeword. Since x is adjacent to t, we are done. $\square$

Theorem 3.6. There is a perfect one error correcting code on Z_n^m for every m and n.

Proof. The above lemmas establish that $C(G_n^m)$ is a perfect one error correcting code on Z_n^m. $\square$

We now establish exactly how many vertices must be codewords in a perfect one error correcting code on Z_n^m.

Definition 3.7. A *1-sphere* of a vertex v in a graph $G = (V, E)$ is a set $\{x \in V | x = v \text{ or } x, v \in E\}$.

We refer to v as the center of the 1-sphere.

Theorem 3.8. If C is a perfect one error correcting code on Z_n^m then

1. if m is odd, $|C| = \frac{n^m+1}{n+1}$ and (Z_n^m, C) has one corner codeword
2. if m is even, $|C| = \frac{n^m+n}{n+1}$ and (Z_n^m, C) has n corner codewords.

Proof. Suppose C is perfect one error correcting code on Z_n^m. Then (Z_n^m, C) is covered with disjoint 1-spheres with codewords as centers. The 1-spheres centered at a corner codeword have n vertices; the other 1-spheres have $n+1$ vertices. Let $A = \{x \in C | x \text{ is a corner codeword}\}$ and let $B = \{x \in C | x \text{ is a non-corner codeword}\}$. Put $i = |A|$ and $j = |B|$. Then

$$n^m = nj + (n+1)i \tag{1}$$

$$n^m \equiv nj \pmod{n+1} \text{ and } 0 \le j \le n \tag{2}$$

Case 1 (m odd): Suppose m is odd. Let r be such that $m = 2r + 1$. From (2), $n^{2r+1} \equiv nj \pmod{n+1}$. Ergo, $n^{2r} \equiv j \pmod{n+1}$. Then, since $n^{2r} \equiv 1 \pmod{n+1}$, $j \equiv 1 \pmod{n+1}$. Since $0 \leq j \leq n$, $n = 1$. From (1) we obtain $i = \frac{n^m - n}{n+1}$. So, $|C| = i + j = \frac{n^m+1}{n+1}$.

Case 2 (m even): Suppose m is even. Let r be such that $m = 2r$. Note that $n^2 \equiv 1 \pmod{n+1}$ as $n^2 = (n+1)(n-1) + 1$. So, $n^{2r} \equiv 1 \equiv n^2 \pmod{n+1}$. From (2), $n^2 \equiv nj \pmod{n+1}$. Hence, $n \equiv j \pmod{n+1}$. Since $0 \leq j \leq n$, $j = n$. From (1), we have $i = \frac{n^m - n^2}{n+1}$. Then $|C| = i + j = \frac{n^m+n}{n+1}$. □

Note that if m is odd we have established there is exactly one corner codeword and if m is even we have established there are exactly n corner codewords.

3.3 Uniqueness

For the following lemmas, let C be a perfect one error correcting code on Z_n^m.

Lemma 3.9. No two blank K_n subgraphs of (Z_n^m, C) are adjacent.

Proof. Suppose $K_n(u)$ and $K_n(v)$ are blank subgraphs of (Z_n^m, C) for some $u, v \in V(U_n^{m-1})$. Let x be such that x joins $K_n(u)$ to $K_n(v)$. Then $\deg x = n$. So, x is a codeword. Hence $K_n(u)$ is not blank. □

Lemma 3.10. Every non-blank K_n subgraph of (Z_n^m, C) is adjacent to at most one blank K_n subgraph.

Proof. Suppose $K_n(u)$ is adjacent to two blank K_n subgraphs of (Z_n^m, C). Let x, y, s, t be such that x joins $K_n(u)$ to blank $K_n(s)$ and y joins $K_n(u)$ to blank $K_n(t)$. Then x, y are codewords. But since $x, y \in K_n(u)$, x is adjacent to y. □

Lemma 3.11. Every K_n subgraph of (Z_n^m, C) not containing a corner vertex is adjeacent to at most one blank K_n subgraph.

Proof. Let $K_n(u)$ be a subgraph of (Z_n^m, C) containing no corner vertex. If $K_n(u)$ were adjacent to no blank K_n subgraph, then x would not join $K_n(u)$ to any blank K_n subgraph. Hence, no $x \in K_n(u)$ would be a codeword. □

Lemma 3.12. There is a K_n subgraph of (Z_n^m, C) containing a corner vertex and not adjacent to a blank K_n subgraph.

Proof. By our above observation, some corner vertex x of (Z_n^m, C) is a codeword. Let v be such that $x \in K_n(v)$. Then $K_n(v)$ is not adjacent to any blank K_n. □

Theorem 3.13. Up to codeword isomorphism, for all m and n there is at most one perfect one error correcting code on Z_n^m and there is at most one weak perfect one error correcting code on Z_n^m.

Proof. Certainly the desired result holds for $m = 1$. Suppose for some $m > 1$, there is at most one weak perfect error correcting code on Z_n^{m-1}. Let $G_1 = (Z_n^m, C_1)$ and $G_2 = (Z_n^m, C_2)$ be such that C_1 and C_2 are perfect one error correcting codes on Z_n^m. Suppose for a moment that G_1 and G_2 are not codeword isomorphic. Consider $\overline{\varphi}$, a graph isomorphism from G_1 to G_2. Suppose that for all blank K_n subgraphs of Z_n^m, $\overline{\varphi}(K_n)$ is blank. Then for any $x \in C_1$, $\overline{\varphi}(x) \in C_2$. This contradicts our supposition that G_1 is not codeword isomorphic to G_2. Hence for all graph isomorphisms $\overline{\varphi}$ from G_1 to G_2, there is a blank K_n subgraph of G_1 such that $\overline{\varphi}(K_n)$ is non-blank. Let $W_1 = \{x \in V(Z_n^{m-1}) | K_n(x)$ is blank in $C_1\}$ and let $W_2 = \{x \in V(Z_n^{m-1}) | K_n(x)$ is blank in $C_2\}$. Let $H_1 = (Z_n^{m-1}, W_1$ and $H_2 = (Z_n^{m-1}, W_2)$. The above lemmas establish that W_1 and W_2 are weak perfect one error correcting codes on Z_n^{m-1}. Now we establish that H_1 and H_2 are not codeword isomorphic. Suppose to the contrary, letting ψ be a codeword isomorphism from H_1 to H_2. Define $\overline{\psi} : V(G_1) \to V(G_2)$ by

1. for non-corner vertex x, $x \mapsto y$ if and only if there are u, v, r, s such that $\psi(r) = s$, $\psi(u) = v$, x joins $K_n(u)$ to $K_n(r)$ and y joins $K_n(v)$ to $K_n(s)$.

2. for corner vertex x, $x \mapsto y$ where y is such that y is the corner vertex in $K_n(\psi(u))$ and u is such that $x \in K_n(u)$.

We claim $\overline{\psi}$ is a graph isomorphism with $\overline{\psi}(K_n)$ blank for all K_n subgraphs of G_1. To that effect, we show that if $\psi(u) = v$ then $\overline{\psi}(K_n(u)) = K_n(v)$. Consider $u \in V(H_1)$ and $v \in V(H_2)$ such that $\psi(u) = v$. Let $y \in K_n(v)$. If y is a corner vertex then clearly y is the image of x, the corner vertex in $K_n(u)$. If y is not a corner verex, y joins $K_n(v)$ to $K_n(w)$ for some $w \in V(H_1)$. Since v is adjacent to w, u is adjacent to $\psi^{-1}(w)$. So, $K_n(u)$ is adjacent to $K_n(\psi^{-1}(w))$. Let x be such that x joins $K_n(u)$ to $K_n(\psi^{-1}(w))$. Then $\overline{\psi}(x) = y$. Hence, for all $y \in K_n(v)$ there is some $x \in K_n(u)$ such that $\overline{\psi}(x) = y$. Then, $\overline{\psi}|_{K_n(u)}$ maps onto $K_n(v)$. Since $|K_n(u)| = |K_n(v)|$ and $|K_n(u)|$ is finite, $\overline{\psi}|_{K_n(u)}$ is a bijection. So, $\overline{\psi}(K_n(u))$=$K_n(v)$.

To see that $\overline{\psi}$ is a graph isomorphism, consider adjacent vertices $x, y \in V(G_1)$. If $x, y \in K_n(u)$ obviously $\overline{\psi}(x)$ is adjacent to $\overline{\psi}(y)$. Suppose $x \in K_n(u)$, $y \in K_n(v)$, $u \neq v$. It follows from the construction of $\overline{\psi}$ that $\overline{\psi}(x)$ is adjacent to $\overline{\psi}(y)$. So $\overline{\psi}$ is a graph isomorphism such that $\overline{\psi}(K_n(u))$ is blank whenever $K_n(u)$ is blank. This contradicts the above that for every graph isomorphism $\overline{\varphi} : G_1 \to G_2$ there is a blank K_n such that $\overline{\varphi}(K_n)$ is non-blank. Hence, our supposition that H_1 is codeword isomorphic to H_2 is false. Hence, H_1 is not codeword isomorphic to H_2. This however contradicts the induction hypothesis. Hence, G_1 is codeword isomorhpic to G_2. A similar argument establish there is at most one perfect error correcting code on Z_n^m. □

Corollary 3.14. For every m and n there is exactly one perfect one error code on Z_n^m.

Proof. Existence was established previously; this together with the previous theorem establishes the claim. □

4 Labeling

Definition 4.1. A *labeled graph* is an ordered triplet $L = (G, S, \pi)$ such that $G = (V, E)$ is a graph and S is a collection of strings over some alphabet and $\pi : V \to S$ is a bijection.

If L is such an ordered triplet, we say L is a labeling on G. By $G(L)$ we mean G and by $S(L)$ we mean S.

Definition 4.2. A *coded labeled graph* is an ordered triplet $L = (H, S, \pi)$ such that $H = (G, C)$ is a coded graph and S is a collection of strings over some alphabet and $\pi : V(H) \to S$ is a bijection.

To simplify our discussion, when we refer to vertex s where $s \in S$, we mean $\pi^{-1}(s)$. When we refer to the first character of a vertex x we mean the leftmost character of $\pi(x)$. Similar to before, we simplify our discussion by referring to graph theoretic properties of L rather than referring to the underlying graph G. The dear reader who has come this far is hopefully clear on this matter by now.

4.1 Labeling Definition

Consider an arbitrary alphabet $\Sigma = \{a_0, a_1, \ldots, a_{n-1}\}$. By Σ_k^* we mean the set of all strings of length k over Σ. For convenience we set $a_0 =$'0'.

We now define a pair of coded labelings $,_n^m$ and Υ_n^m on Z_n^m.

Let $\pi_n^1 : V(Z_n^1) \to \Sigma_1^*$ be any bijection. Let $C_n^1 = \{x \in V(Z_n^1) | \pi(x) = a_0\}$. Let $H_n^1 = (Z_n^1, C_n^1)$. Put $,_n^1 = (H_n^1, \Sigma_1^*, \pi_n^1)$. To construct $,_n^m$ for $m > 1$ and m even, form n (coded labeled) copies $C_0, C_1, \ldots, C_{n-1}$ of $,_n^{m-1}$. For $i \neq j$, form exactly one edge between C_i and C_j such that

1. the edge is incident on corner vertices $x \in V(C_i)$ and $y \in V(C_j)$
2. the leftmost character of x is a_k where $k \equiv j - i \pmod{n}$
3. the leftmost character of y is a_k where $k \equiv i - j \pmod{n}$.

Lastly, we must define $\pi_n^m : V(Z_n^m) \to \Sigma_m^*$. For $x \in V(C_i), x \mapsto a_i \pi_n^{m-1}(x)$. That this process results in $G(,_n^m) = Z_n^m$ and $S(,_n^m) = \Sigma_m^*$ is clear.

To construct $,_n^m$ for $m > 1$ and m odd, form one copy C_0 of $,_n^{m-1}$ and $n - 1$ copies $C_1, C_2, \ldots, C_{n-1}$ of Υ_n^{m-1}. For $i \neq j$, form exactly one edge between C_i and C_j such that

1. the edge is incident on corner vertices $x \in V(C_i)$ and $y \in V(C_j)$
2. the leftmost character of x is a_j
3. the leftmost character of y is a_i.

Lastly, we must define $\pi_n^m : V(Z_n^m) \to \Sigma_m^*$. For $x \in V(C_i), x \mapsto a_i \pi_n^{m-1}(x)$. That this process results in $G(,{}_n^m) = Z_n^m$ and $S(,{}_n^m) = \Sigma_m^*$ is clear.

Let $\rho_n^1 : V(Z_n^1) \to \Sigma_1^*$ be any bijection. Let $C_n^1 = \emptyset$. Let $K_n^1 = (Z_n^1, C_n^1)$. Put $\Upsilon_n^1 = (K_n^1, \Sigma_1^*, \rho_n^1)$. To construct Υ_n^m for $m > 1$ and m odd, form n copies $C_0, C_1, \ldots, C_{n-1}$ of Υ_n^{m-1}. For $i \neq j$, form exactly one edge between C_i and C_j such that

1. the edge is incident on corner vertices $x \in V(C_i)$ and $y \in V(C_j)$
2. the leftmost character of x is a_k where $k \equiv j - i \pmod{n}$
3. the leftmost character of y is a_k where $k \equiv i - j \pmod{n}$.

Lastly, we must define $\rho_n^m : V(Z_n^m) \to \Sigma_m^*$. For $x \in V(C_i), x \mapsto a_i \rho_n^{m-1}(x)$. That this process results in $G(\Upsilon_n^m) = Z_n^m$ and $S(\Upsilon_n^m) = \Sigma_m^*$ is clear.

To construct Υ_n^m for $m > 1$ and m even, form one copy C_0 of Υ_n^{m-1} and $n-1$ copies $C_1, C_2, \ldots, C_{n-1}$ of $,{}_n^{m-1}$ For $i \neq j$, form exactly one edge between C_i and C_j such that

1. the edge is incident on corner vertices $x \in V(C_i)$ and $y \in V(C_j)$
2. the leftmost character of x is a_j
3. the leftmost character of y is a_i.

Lastly, we must define $\rho_n^m : V(Z_n^m) \to \Sigma_m^*$. For $x \in V(C_i), x \mapsto a_i \rho_n^{m-1}(x)$. That this process results in $G(\Upsilon_n^m) = Z_n^m$ and $S(\Upsilon_n^m) = \Sigma_m^*$ is clear.

Note there is exactly one corner vertex in $,{}_n^m$ with leftmost digit a_i and there is exactly one corner vertex in Υ_n^m with leftmost digit a_i.

We now want to establish that the above construction defines a perfect one error correcting code on Z_n^m.

Lemma 4.3. For any m and n, the following hold:

1. for any vertex v of $,{}_n^m$, v is a corner codeword if and only if m is odd and $v = \underbrace{0\cdots0}_{m}$ or m even and $v = a_i\underbrace{0\cdots0}_{m-1}$ where $0 \le i < n$
2. for any corner vertex v of Υ_n^m, v is a non-codeword not adjacent to a codeword if and only if m even and $v = \underbrace{0\cdots0}_{m}$ or m odd and $v = a_i\underbrace{0\cdots0}_{m-1}$ where $0 \le i < n$
3. no corner vertex of Υ_n^m is a codeword.

Proof. The statement clearly holds for $m = 1$ and $m = 2$. Suppose the desired claim holds for $m - 1$ where $m > 1$.

1. **Case 1 (m even):** Here we handle the case m even. Consider a corner codeword $v \in V(,{}_n^m)$. Suppose v has label $a_i s$ where $s \in \Sigma_{m-1}^*$. Since v is a corner codeword, s must be the label of a corner codeword w in the copy C_i of $,{}_n^{m-1}$. As a consquence of $m-1$ being odd, $s = \underbrace{0\cdots0}_{m-1}$. So, w has label $a_i\underbrace{0\cdots0}_{m-1}$. From our above observation that the perfect one error correcting code on Z_n^m has n corner codewords, it follows there is such a w with label $a_i\underbrace{0\cdots0}_{m-1}$ for each $0 \leq i < n$.

   Case 2 (m odd): Consider a corner codeword $v \in V(,{}_n^m)$. Note that v must be a vertex in the copy C_0 of $,{}_n^{m-1}$ since no corner in a copy of Υ_n^{m-1} is a codeword. So, v has a label $0s$ where $s \in \Sigma_{m-1}^*$. Note v connects C_0 to a copy C_i of Υ_n^{m-1} if and only if the leftmost character of s is a_i. So, v is the corner in $,{}_n^m$ if and only if the leftmost character of s is '0'. From above, $\underbrace{0\cdots0}_{m-1}$ is a corner codeword in $,{}_n^{m-1}$. Since the leftmost digit of s is '0', $s = \underbrace{0\cdots0}_{m-1}$. Hence, $v = \underbrace{0\cdots0}_{m}$.

2. Here we handle the case m odd. We omit the case m even as the argument is similar. Consider a non-codeword corner vertex v in $V(\Upsilon_n^m)$ which is adjacent to no codeword. Suppose v has label $a_i s$ where $s \in \Sigma_{m-1}^*$ Note the label s corresponds to a vertex in Υ_n^{m-1} that is a corner non-codeword not adjacent to a codeword. Since $m-1$ is even, $s = \underbrace{0\cdots0}_{m-1}$. So, v has label $a_i\underbrace{0\cdots0}_{m-1}$. By our above lemma, since Υ_n^m has n corner non-codewords there is such a v for each $0 \leq i < n$.

3. **Case 1 (m odd):** Each corner vertex in Υ_n^m is a corner in a copy of Υ_n^{m-1} and no corner is a codeword there.

   Case 2 (m even): Since no corners of Υ_n^{m-1} are codewords and only $\underbrace{0\cdots0}_{m-1}$ is a corner codeword in $,{}_n^{m-1}$, we need only ensure that each vertex $a_i\underbrace{0\cdots0}_{m-1}$ in a copy C_i of $,{}_n^{m-1}$ joins C_i to some distinct copy C_j. Since the first digit of $\underbrace{0\cdots0}_{m-1}$ is '0', $a_i\underbrace{0\cdots0}_{m-1}$ joins C_i to C_0.

$\square$

Theorem 4.4. For all m and n:

1. in $,{}_n^m$, every non-codeword is adjacent to at least one codeword; in Υ_n^m, every non-corner non-codeword is adjacent to at least one codeword

2. in , $^{m}_{n}$ and Υ^m_n, every non-codeword is adjacent to at most one codeword

3. in , $^{m}_{n}$ and Υ^m_n, no two codewords are adjacent.

Proof. Clearly the desired results holds for $m = 1$ and $m = 2$; suppose the claim holds for $m - 1$ for some $m > 1$. We prove that the above hold for , $^{m}_{n}$; similar arguments apply to Υ^m_n. The unconvinced reader may supply them if he wishes.

Case 1 (m even): Since every non-codeword in the copies of , $^{m-1}_{n}$ are adjacent to some codeword, every non-codeword in , $^{m}_{n}$ is adjacent to some codeword.

Since 2 and 3 hold for the copies of , $^{m-1}_{n}$, to establish 2 and 3 for , $^{m}_{n}$, we need only notice that for any new edge $\{x, y\}$ between distinct copies C_i and C_j of , $^{m-1}_{n}$, neither x nor y is a codeword. Were x a codeword, by a previous lemma, $x = \underbrace{0 \cdots 0}_{m}$. But then $0 \equiv i - j \pmod{n}$, contradicting $i \neq j$.

Case 2 (m odd): We need only consider corners of the copies of Υ^{m-1}_n not adjacent to any codeword in their respective copy of Υ^{m-1}_n. By a previous lemma, $a_i \underbrace{0 \cdots 0}_{m-1}$ is the only such corner in C_i, a copy of Υ^{m-1}_n. Since n is odd, $n - 1$ is even. So, every corner of , $^{m-1}_{n}$ is a codeword. Consider an edge $\{x, a_i t\}$ from C_0, the copy of , $^{m-1}_{n}$ to C_i. By construction, the leftmost character of t must be 0. Since there is only one corner vertex whose leftmost character is 0, t must be $\underbrace{0 \cdots 0}_{m-1}$ since $\underbrace{0 \cdots 0}_{m-1}$ is a corner in Υ^{m-1}_n. Hence $a_i \underbrace{0 \cdots 0}_{m-1}$ is adjacent to x, a codeword.

Suppose there is a non-codeword adjacent to two codewords. Let C_i be the copy of Υ^{m-1}_n such that there is a $v \in V(\Upsilon^{m-1}_n)$ which is adjacent to two codewords. One codeword must be in a copy C_i of Υ^{m-1}_n, the other must be in the copy C_0 of , $^{m-1}_{n}$. Consider the edge $x, a_i t$ from C_0 to Ci. Our immediately preceeding argument for 1 showed that $t = \underbrace{0 \cdots 0}_{m-1}$. Hence, $v = a_i \underbrace{0 \cdots 0}_{m-1}$ and so v is adjacent to no codeword in Υ^{m-1}_n. This establishes 2.

Clearly 3 holds, as no corners of the copies of Υ^{m-1}_n are codewords and every new edge formed between distinct copies must contain a vertex in a copy of Υ^{m-1}_n. $\square$

This proof shows that the construction method yields a perfect one error correcting on Z^m_n. By uniqueness then, it follows that

Corollary 4.5. For all m and n $C(G^m_n) = C(, ^{m}_{n})$.

Proof. Obvious. $\square$

4.2 Codeword Characterization

Keep in mind that throughout we have made numerous simplifications of phrasing. For example, if we say vertex v ends in an odd number of zeros we mean $\pi(v)$ ends in an odd number of zeros. If we say vertex $v = a_{i_0} a_{i_1} \cdots a_{i_{n-1}}$ we mean $\pi(v) = a_{i_0} a_{i_1} \cdots a_{i_{n-1}}$. Other such simplifications are clear from context and we will not dwell on the matter any further. We state a characterization of the codewords in $,{}^m_n$.

Theorem 4.6. For all m and n, the following hold:

1. for all vertices v of $,{}^m_n$, v is a codeword if and only if v ends in an odd number of zeros or $v = \underbrace{0 \cdots 0}_{m}$
2. for all vertices v of Υ^m_n, v is a codeword if and only if v ends in an odd number of zeros and $v \neq \underbrace{0 \cdots 0}_{m}$.

Proof. Obviously the desired result holds for $m = 1$. Suppose the claim holds for $m - 1$ where $m > 1$. We use this to establish the desired result for m. Suppose m is odd. Consider $v \in V(,{}^m_n)$. Suppose v has label $a_i s$ where $s \in \Sigma^*_{m-1}$.

If v is in the one copy C_0 of $,{}^{m-1}_n$, then $a_i = 0$. Note v ends in an odd number of zeros or $v = \underbrace{0 \cdots 0}_{m}$ if and only if the string s ends in an odd number of zeros or $s = \underbrace{0 \cdots 0}_{m-1}$. That is, if and only if s is the label of a codeword in $,{}^{m-1}_n$. That is, if and only if v is a codeword.

Suppose v is in a copy C_i of Υ^{m-1}_n. Then, $a_i \neq 0$. Note v ends in an odd number of zeros or $v = \underbrace{0 \cdots 0}_{m}$ if and only if the string s ends in an odd number of zeros or $s = \underbrace{0 \cdots 0}_{m-1}$. That is, if and only if s is the label of a codeword in $,{}^{m-1}_n$. That is, if and only if v is a codeword.

The remaining cases are equally trivial and we omit them for sake of brevity. □

The codeword recognizor finite state machine works by scanning strings over Σ from right to left. 'All' refers to any character and 'Nonzero' refers to any of $a_1, a_2, \ldots, a_{n-1}$. The following observations are trivial to verify.

1. if $w = \underbrace{0 \cdots 0}_{m}$, then the codeword recgonizer will end in state $S1$ or $S2$ given w as input
2. if $w \neq \underbrace{0 \cdots 0}_{m}$ ends in an odd number of zeros, then the codeword recognizor will end in state $S3$ given w as input

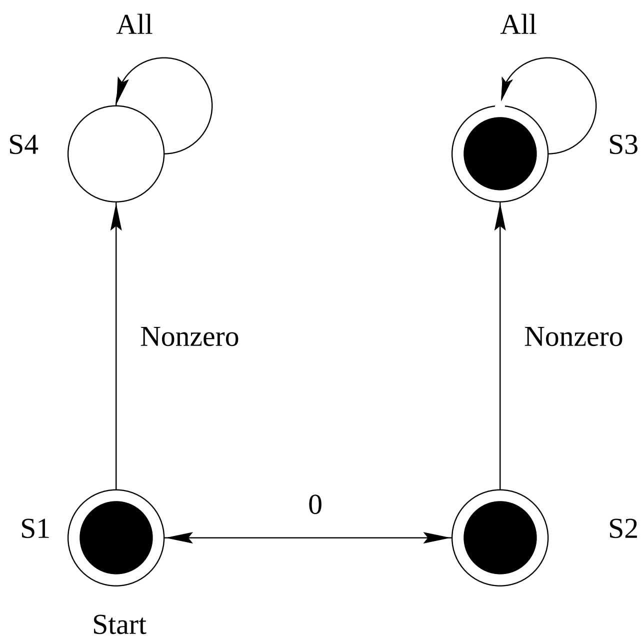

Figure 7: Codeword Recognizor

3. if $w \neq \underbrace{0 \cdots 0}_{m}$ ends in an even number of zeros, then the codeword recognizor will end in state $S4$ given w as input.

Hence, the codeword recognizor will end in $S4$ if and only if w is not a codeword.

4.3 Error Correction

Definition 4.7. A non-codeword is of *type*

1. R if it ends in a zero
2. E if it ends in a nonzero preceeded by an even number of zeros
3. L if it ends in a nonzero preceeded by an odd number of zeros.

The finite state machine which sorts strings into these is shown as figure 8. The machine reads input strings of length at least two from right to left.

Define the following unary operations R, E and L on s a string over Σ as:

1. $R(s)$ is swap the positions of the first nonzero character from right of s with character to its right
2. $E(s)$ is change the rightmost character of s to '0'

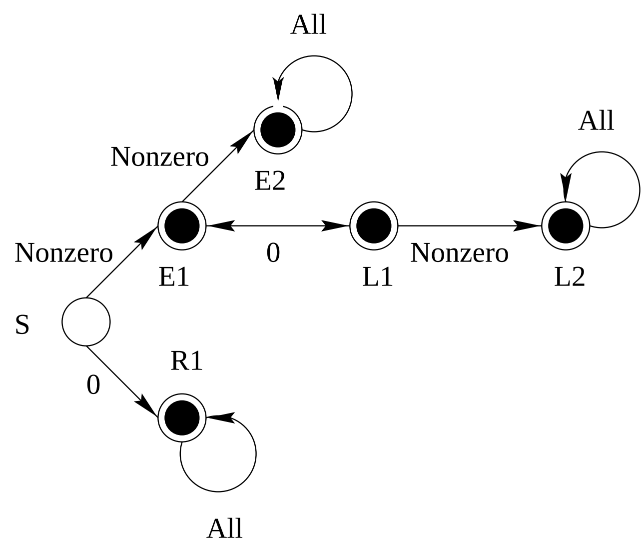

Figure 8: String sorter

3. $L(s)$ is swap the positions of the first nonzero character from right of s with character to its left.

The finite state machine which sorts strings into the types R, L and E is shown as figure 8. The machine reads input strings from right to left. Consider the input string s, a non-codeword adjacent to codeword t. If s causes the machine to halt in $R1$ then $R(s) = t$; if s causes the machine to halt in $L1$ or $L2$ then $L(s) = t$; if s causes the machine to halt in $E1$ or $E2$ then $E(s) = t$. As is justified by the following theorem, this can be readily verified by checking that strings of type T cause the machine to halt in a state Ti, where T is one of R, L or E.

Theorem 4.8. For all m and n and non-codeword vertex x adjacent to codeword vertex y in , ^m_n or Υ^m_n, if x is of type T then $T(x) = y$ where T is one of R, E or L.

Proof. Clearly the desired result holds for $m = 1$. Suppose $m > 1$ and consider v a non-codeword in a copy C_i of , $^{m-1}_n$ or Υ^{m-1}_n. Then v has label $a_i s$ where $s \in \Sigma^*_{m-1}$. Since prefixing a string will preserve the type, if v is adjacent to a codeword in C_i, the same operation will correct v that will correct the vertex with label s in the copy C_i. So we need only consider those non-codewords w adjacent to a codeword in C_j for $i \neq j$. We prove the claim for , $^{m-1}_n$. The claim is establish similarly for Υ^{m-1}_n.

Case 1 (m even): For m even, every non-codeword in a copy of , $^{m-1}_n$. Since , $^{m-1}_n$ is a coded labeled Z^{m-1}_n such that the code is a perfect one error correcting code, every non-codeword in , $^{m-1}_n$ is adjacent to some codeword in that copy. So, , ${}^{m}_{n}$ contains no codewords adjacent to some codeword in a distinct copy of , ${}^{m-1}_{n}$.

Case 2 (m odd): Every codeword in the copy C_0 of , ${}^{m-1}_{n}$ is adjacent to some codeword in that copy as , ${}^{m-1}_{n}$ carries with it a perfect one error correcting code on Z_n^{m-1}. So any non-codeword v adjacent to a codeword in a distinct copy must occur in a copy C_i of Υ_n^{m-1} and be adjacent to a codeword w in the copy C_0 of , ${}^{m-1}_{n}$. So, $v = a_i s$ where $s \in \Sigma^*_{m-1}$ is the label of a corner vertex adjacent to no codewords in Υ_n^{m-1}. Since $m-1$ is even, by a previous lemma, $s = \underbrace{0 \cdots 0}_{m-1}$. Then $v = a_i \underbrace{0 \cdots 0}_{m-1}$. Now consider w, the codeword adjacent to v. Note $w = 0t$ where $t \in \Sigma^*_{m-1}$ and t is the label of a corner codeword in , ${}^{m-1}_{n}$. Since $m-1$ is even, by a previous lemma, $t = a_k \underbrace{0 \cdots 0}_{m-2}$ for some $0 \leq k < n$. Since an edge connects t in the copy C_0 of , ${}^{m-1}_{n}$ and the vertex with label s in the copy C_i, $i = j$. So, $w = 0a_i \underbrace{0 \cdots 0}_{m-2}$. Hence, v is of type R and $R(v) = w$. □

4.4 Nonlinearity

In the following discussion it is convenient to think of the set of codewords on , ${}^{m}_{n}$ as the labels which have been assigned to the codewords rather than the vertices.

A code C is said to be linear if C is a subspace of the vector space V^n which is the set of n component vectors over some set of scalars V. If V^n with an appropriate operation is a group, then C is linear if and only if C is a subgroup of V^n.

From Lagrange's theorem, the order of a subgroup must divide the order of a group. In our case, $V^m = \Sigma^*_m$. Note $|V^m| = n^m$. From our counting argument we can see that $|C|$ will not divide $|V^m|$ except in a few special cases. In fact,

Theorem 4.9. If $m > 2$ than the perfect one error correcting code on Z_n^m is nonlinear.

Proof. Suppose $m > 2$. If m odd, then the number of codewords on Z_n^m is $\frac{n^m+1}{n+1}$ which doesn't divide n^m. If m even, then the number of codewords on Z_n^m is $\frac{n^m+n}{n+1}$ which doesn't divide n^m. □

5 Conclusion

We have presented a biinfinite family of graphs and demonstrated that on these graphs there is a unique perfect one error correcting code. We showed how to label these graphs so that recognition of codewords and error correction is not hard.

6 Bibliography

[CN99] P. Cull and I. Nelson. Perfect Codes, NP-Completeness, and Towers of Hanoi Graphs. *Bulletin of the Institute of Combinatorics and its Applications*, 1999.