Maximal and Reversible Snakes in Hypercubes

Dayanand S. Rajan, Goldman Sachs & Co., USA. *email*: dayanand.rajan@gs.com

Anil M. Shende, Mathematics, Computer Science and Physics Department, Roanoke College, USA.
email: shende@roanoke.edu

1 Introduction

The problem of finding the length of a longest *snake*, that is, an induced path, in a hypercube was posed in 1958 [Kau58] in the context of coding theory, and is still an open problem. Subsequently this problem has been shown to be of importance in connection with several other applied problems, e.g., electronic combination locking schemes (see, for example, [Bla64, CFT64, Kle70, Jab74, PT96] for discussions on some of these applications).

Over the last couple of years we have carefully analysed known longest d-snakes for $d <= 7$. This analysis has led to a host of possibilities for further investigation. In this paper we address two of these possibilities that seem the most promising. Neither of the two approaches we mention below have been studied in the literature.

2 Notation and Definitions

A *snake in a d-dimensional hypercube* is an *induced path* in the d-dimensional hypercubical, undirected graph. The *length* of a snake is, by definition, the number of edges in the snake. Longest snakes in three and four dimensional hypercubes are shown in Figures 1 and 2, respectively.

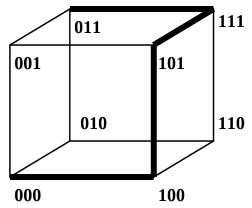

Figure 1: A Longest Snake in a 3-dimensional Cube

A related notion is that of a *coil in a d-dimensional hypercube*, which is, by definition, an *induced cycle* in the corresponding d-dimensional hypercubical, undirected graph (see also [AK88, HHW88]). Unfortunately, the literature is not consistent on the use of terminology for these two notions. What we have defined as a snake is sometimes called an *open snake* in the literature, and what we have defined as a *coil* is sometimes referred to as a *snake* or a *closed snake* in the literature.

In this paper we are concerned *only* with *snakes* as defined above. For brevity, we will refer to a snake (coil) in a d-dimensional hypercube by a *d-snake* (*d-coil*).

We will use the following notation. For each $d > 0$, we will denote the set $\{0, 1, \ldots, (d-1)\}$ by $\mathbf{d}$. $\mathcal{P}^{\mathbf{d}}$ will denote the set of all subsets of $\mathbf{d}$, and $\mathbf{2}^{\mathbf{d}}$ will denote the set of characteristic functions of $\mathcal{P}^{\mathbf{d}}$. A *d-dimensional hypercubical graph* (or simply *d-dimensional hypercube*), denoted by H^d, is, by definition, an undirected graph with 2^d vertices, so that

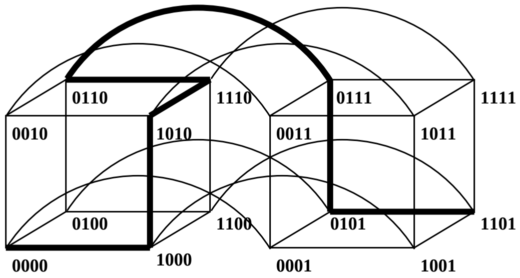

Figure 2: A Longest Snake in a 4-dimensional Hypercube

1. each vertex is labelled with the characteristic function for a unique subset of $\mathbf{d}$, and

2. for any two vertices, v and w, with labels f_v and f_w respectively, (v, w) is an edge in H^d (i.e., v and w are neighbours in H^d) iff f_v and f_w differ in exactly one place.

Henceforth, we will not make any distinction between vertices in H^d and the labels assigned to them. Figures 1 and 2 show labelled 3 and 4 dimensional cubes.

Then, by definition, a d-snake, S_d, is a sequence, $(s_0, s_1, \ldots, s_k)$, of elements of $\mathbf{2^d}$, such that for all i, j, $0 \leq i < j < k$, s_i and s_j are neighbours in H^d iff $j = i + 1$. The length of this snake is k, the number of edges of H^d in the snake.

$s(d)$ denotes the length of a *longest d-snake.* (Similarly, $c(d)$ denotes the length of a *longest d-coil.*)

Without loss of generality, we can assume that every d-snake $S_d = (s_0, s_1, \ldots, s_k)$, is such that $s_0 = \emptyset$. Then, we will represent S_d by a string $c_0 c_1 \ldots c_{k-1}$ over $\mathbf{d}$ so that for each $i, 0 \leq i \leq (k-1), s_i(c_i) \neq s_{(i+1)}(c_i)$, that is, the labels s_i and $s_{(i+1)}$ differ in the c_i-th place. Thus,

$$S_d = (s_0 = \emptyset, s_1, \ldots, s_k) = c_0 c_1 \ldots c_{k-1}.$$

3 A Brief Literature Survey

The problems of finding values for $s(d)$ and $c(d)$ for each $d > 0$ are both open. It is known that the problem of finding the length of a longest induced path in a graph is NP-complete, and so is the problem when the graphs are restricted to bipartite graphs [GJ79]. To our knowledge, it is not known whether the problem, further restricted to hypercubical graphs, is NP-complete.

For $d >= 7$, the best known lower and upper bounds for $c(d)$ are (see [AK91, Sne94])

$$\frac{77}{256} 2^d <= c(d) <= 2^{d-1} \left(1 - \frac{1}{(d^2 - d + 2)}\right).$$

This leaves a large margin for error: $\Theta(2^d/d^2)$. (For $d >= 12$, a better upper bound of $2^{d-1}(1 - 1/(20n - 41))$ is known, but it does not drastically reduce the margin for error.)

Since, $s(d) >= c(d) - 2$, the best known lower bound for $s(d)$ is given by $(77/256) \cdot 2^d - 2 <= s(d)$. The exact values for $s(d)$ are known, using an exhaustive search algorithm, for $d <= 7$. The best known value of $c(8)$ is 96, and thus $s(d) >= 94$ (see [PT96]).

Dim. d	Longest d-snake
3	**01**20
4	**0120**310
5	**012301**4312301
6	**012310430**54013402410431534
7	**0123453210325312345231032**6054301350231035430125023

Table 1: Maximal $(d-1)$-snakes as initial segments of longest d-snakes.

4 Maximal Snakes

Definition 1 *A snake $S_d = (s_0, s_1, \ldots, s_k)$ is* maximal *in H^d iff, by definition, there is no vertex v in H^d such that $(v, s_0, \ldots, s_k)$ or $(s_0, \ldots, s_k, v)$ is a snake in H^d.*

Clearly, then, if $S_d = (s_0, s_1, \ldots, s_k)$ is a *maximal* d-snake then so is ${S'}_d = (s_k, s_{k-1}, \ldots, s_0)$.

We noticed that for each $d <= 7$, there is a longest d-snake that has a *maximal* $(d-1)$-snake, not necessarily a longest $(d-1)$-snake, as its initial segment. Table 1 shows longest d-snakes with the maximal $(d-1)$-snake initial segments in bold face.

Conjecturing, then, that for *all* d, there is a longest d-snake that has a maximal $(d-1)$-snake as its initial segment, we focused our attention on studying *all* maximal d-snakes (not just the longest ones), and the extension of maximal d-snakes to maximal $(d+1)$-snakes. While experimenting with this conjecture, we generated a maximal 8-snake of length 97, thus *improving* the best known lower bound for $s(8)$. Below is the length 97 8-snake, with the maximal, length 50, 7-snake shown in bold face.

01234532103253123452310326054301350231035430125023
76321035430135023103543013624351035431241035431

Note that, in this case, the maximal 7-snake is also a longest 7-snake. Note also that the reversal of the above length 97 8-snake, has as its initial segment a length 46 maximal 7-snake.

Our study of maximal d-snakes also led us to conjecture that

Conjecture 1 *For each d, for each k, $2(d-1) <= k <= s(d)$, there is a maximal d-snake of length k.*

We have verified this conjecture for $d <= 6$. We have now found that there is no maximal 7-snake of length 13, thus disproving our conjecture in general. This observation opens up a new problem of characterizing the lengths for which maximal d-snakes do exist.

In Theorem 1 we show that $2(d-1)$ is a lower bound on the length of maximal d-snakes. We then show (Theorem 2) that this is also a tight lower bound.

The following lemma follows immediately from the definition of maximal snakes.

Lemma 1 *Suppose $S_d = c_0 \ldots c_{(k-1)}$ is a maximal d-snake. Then, $\{c_i | 0 \leq i < k\} = \mathbf{d}$.* $\square$

It is useful to introduce the following notation at this point. For each $n \leq d$, let

$$X^{(d,n)} = \{c | (c \in \mathcal{P}^{\mathbf{d}}) \text{ and } (|c| = n)\}.$$

Theorem 1 *For each $d > 0$, for each $k < 2(d-1)$, there is no maximal d-snake of length k.*

PROOF. The statement of the theorem is easily verifiable for $d <= 3$. Let $d > 3$. Suppose, by way of contradiction,

$$S_d = c_0 \ldots c_{(k-1)} = (\emptyset, s_1, \ldots, s_k)$$

is a maximal d-snake and $k < 2(d-1)$. By Lemma 1 $\{c_i | 0 \leq i < k\} = \mathbf{d}$. Since the first vertex of S_d is $\emptyset$, and since S_d is a maximal snake, s_1 is the only vertex on the snake that is in $X^{(d,1)}$. Thus, for each $i, 2 \leq i \leq k$, $s_i \in X^{(d,n)}$, where $n \geq 2$, and $s_2 \in X^{(d,2)}$. Let $s_{e_0} = s_2, s_{e_1}, \ldots, s_{e_m}$ be the sequence of vertices along S_d such that for each $i, 0 \leq i \leq m$, $s_{e_i} \in X^{(d,2)}$. Essentially, this sequence of vertices consists of all those vertices on S_d that are at a distance (measured as the path length of a shortest path in H^d) of 2 from the vertex $\emptyset$, and hence each of these two vertices is a subset of $\mathbf{d}$ containing two elements. Intuitively, each of these vertices "covers" two of the dimensions in $\mathbf{d}$. We will proceed to show that over all the vertices in this sequence, at most $d-1$ dimensions can be "covered", and hence, the snake can be extended at the vertex $\emptyset$.

If for any i, $e_{i+1} - e_i = 2$, then there is a t such that $s_{e_i} \cap s_{e_{(i+1)}} = \{t\}$. Define g so that for each $i, 1 \leq i \leq m$,

$$\begin{aligned} g(i) &= 1, \text{ if } e_i - e_{(i-1)} = 2, \\ &= 2, \text{ otherwise.} \end{aligned}$$

Essentially, $g(i)$ is the maximum number of dimensions that are "covered" by s_{e_i} in addition to the ones "covered" by $s_{e_{(i-1)}}$.

Let N denote the cardinality of the set

$$E = \bigcup_{i=0}^{m} s_{e_i}.$$

That is, N is the actual number of dimensions "covered" by those vertices of S_d that are at distance 2 from the vertex $\emptyset$. Then, since $s_2 = s_{e_0}$ can cover at most 2 dimensions,

$$N \leq 2 + \sum_{i=1}^{m} g(i). \tag{1}$$

Let us now count the number of vertices, V, of S_d between s_3 and s_{e_m}, both inclusive. For each $i, 1 \leq i \leq m$, let $v(i)$ denote the number of vertices between $s_{e_{(i-1)}}$ and s_{e_i}, including s_{e_i}. Then,

$$V = \sum_{i=1}^{m} v(i).$$

Note that $v(i) = 2$ if $g(i) = 1$, and $v(i) \geq 4$ if $g(i) = 2$. Therefore, for each $i, 1 \leq i \leq m$, $v(i) \geq 2 \cdot g(i)$. Therefore,

$$2 \cdot \sum_{i=1}^{m} g(i) \leq \sum_{i=1}^{m} v(i) = V. \tag{2}$$

Since the total number of vertices on S_d are $(k+1)$, and since V is the vertex count starting at c_3, $V \leq (k+1-3)$. Therefore, by Equation (2)

$$\sum_{i=1}^{m} g(i) \leq \frac{(k+1-3)}{2}. \tag{3}$$

Then, using Equations (1) and (3), and the premise that $k < 2(d-1)$, we have

$$\begin{aligned} N &\leq 2 + \sum_{i=1}^{m} g(i) \\ &\leq 2 + \frac{(k+1-3)}{2} \\ &= \frac{k+2}{2} \\ &< \frac{2(d-1)+2}{2} \\ &= d. \end{aligned}$$

Thus we have that $N < d$. Therefore, there is a $t \in \mathbf{d}$, such that $t \notin E$. That is, there is at least one dimension t not "covered" by any of the vertices in the sequence of vertices on S_d that are at a distance of 2 from $\emptyset$. Then, S_d can be extended at $\emptyset$ with the vertex $\{t\}$, and thus is not a maximal snake, giving us a contradiction. □

We will use the following lemma in the proof of Theorem 2. For each d, for each $t \in \mathbf{d}$, let

$$C^{(d,2,t)} = \{c \in X^{(d,2)} | t \in c\}.$$

That is, $C^{(d,2,t)}$ is the set of all vertices of H^d that are at distance 2 from $\emptyset$ and that contain t.

Lemma 2 *Suppose $S_d = (s_0 = \emptyset, s_1, \ldots, s_k)$ is a d-snake. If there is a $t \in \mathbf{d}$ such that (1) $C^{(d,2,t)} \subseteq \{s_i | 0 \leq i \leq k\}$ and (2) $s_k \in C^{(d,2,t)}$, then S_d is a maximal d-snake.*

PROOF. Suppose the premise of the lemma. Then, each vertex in $X^{(d,1)}$ is either on S_d or has a neighbour on S_d. Thus, S_d cannot be extended from $\emptyset$. Since for some $t \in \mathbf{d}$, $s_k \in C^{(d,2,t)}$, there is a $j \in \mathbf{d}$ such that $s_k = \{t, j\}$. The set of neighbours of s_k is

$$\{\{t\}, \{j\}\} \cup \{\{t, j, m\} | m \in (\mathbf{d} \backslash \{t, j\})\}.$$

Both of $\{t\}$ and $\{j\}$ are neighbours of $\emptyset$. Moreover, since each element of $C^{(d,2,t)}$ is on S_d, each $\{t, j, m\}, m \in (\mathbf{d} \backslash \{t, j\})$ has two neighbours on S_d, namely, $\{t, j\}$ and $\{t, m\}$. Therefore, none of the neighbours can extend S_d from s_k, thus proving the lemma. □

Theorem 2 *For each $d > 1$, there is a maximal d-snake of length $2(d-1)$.*

PROOF. It is easily verified that for $d = 2, 3$, the d-snakes 01 and 0120, respectively, are maximal d-snakes.

We will now prove by induction that for each $d > 3$, the d-snake

$$012032 \ldots (d-1)(d-2)$$

satisfies conditions (1) and (2) of Lemma 2 for $t = 1$, and thus is a maximal d-snake.
Base case: $d = 4$. We need to show that the 4-snake $S_4 = 012032$ satisfies conditions (1) and (2) of Lemma 2 for $t = 1$. As a sequence of vertices,

$$S_4 = (s_0 = \emptyset, s_1 = \{0\}, s_2 = \{0, 1\}, s_3 = \{0, 1, 2\}, s_4 = \{1, 2\}, s_5 = \{1, 2, 3\}, s_6 = \{1, 3\}).$$

Thus, $C^{(d,2,1)} \subseteq \{s_i | 0 \leq i \leq 6\}$ and $s_6 = \{1, d-1\} \in C^{(d,2,1)}$.
Induction Hypothesis: For some $p \geq 4$, the p-snake $S_p = 012032\ldots(p-1)(p-2) = (s_0 = \emptyset, s_1, s_2, \ldots, s_k)$ satisfies the conditions:

1. $C^{(p,2,1)} \subseteq \{s_i | 0 \leq i \leq k\}$, and
2. $s_k = \{1,(p-1)\} \in C^{(p,2,1)}$.

To show that:

$$\begin{aligned} S_{p+1} &= 01203243\ldots(p-1)(p-2)(p+1-1)(p+1-2) \\ &= (u_0 = \emptyset, u_1, \ldots, u_k, u_{(k+1)}, u_{(k+2)} \end{aligned}$$

satisfies the conditions:

1. $C^{((p+1),2,1)} \subseteq \{u_i | 0 \leq i \leq (k+2)\}$, and
2. $u_{(k+2)} = \{1,(p+1-1)\} \in C^{((p+1),2,1)}$.

We first note that for each $i, 0 \leq i \leq k$, as sets, $s_i = u_i$, but as characteristic functions the domain of u_i is an extension, by p, of the domain of s_i. Thus, from condition (2) of the IH, $u_k = \{1,(p-1)\}$. Therefore, using the representation for the chosen $S_{(p+1)}$, we have $u_{(k+1)} = \{1,(p-1),p\}$ and $u_{(k+2)} = \{1,p\} \in C^{((p+1),2,1)}$, thus showing that the chosen $S_{(p+1)}$ satisfies condition (2) above.

Also it follows from our note just above that for each $i, 0 \leq i \leq k$, $s_i \in C^{(p,2,1)}$ iff $u_i \in C^{((p+1),2,1)}$. Then, by condition (1) of the IH,

$$C^{((p+1),2,1)} \backslash \{\{1,p\}\} \subseteq \{u_i | 0 \leq i \leq k\}.$$

Moreover, as shown above, $u_{(k+2)} = \{1,p\}$, thus showing that the chosen $S_{(p+1)}$ satisfies condition (1) above.

Therefore, we have that for each $d > 3$, $S_d = 012032\ldots(d-1)(d-2)$ satisfies conditions (1) and (2) of Lemma 2 for $t = 1$, and thus is a maximal d-snake of length $2(d-1)$. □

5 Reversible Snakes

Another interesting observation we have is that for d even, $d <= 6$, there is a *unique* (up to isomorphism) longest d-snake. Consequently, this snake is a reversal of itself as witnessed by a suitable permutation of $\mathbf{d}$. Moreover, this snake contains, as sub-snakes, two isomorphic copies of the unique longest $(d-2)$-snake, This led to our study of the permutations of $\mathbf{d}$ with respect to their use in the generation of maximal, reversible d-snakes. We present an algorithm and prove its correctness to show that we can generate *exactly one* d-snake from each isomorphism class of maximal, reversible d-snakes by considering *only* $\lfloor(d/2)\rfloor$ of the $d!$ permutations of $\mathbf{d}$.

Henceforth, we will assume that all d-snakes begin at the vertex $\emptyset$.

Definition 2

1. *Two d-snakes $S_d = c_0c_1\ldots c_{(k-1)}$ and $S'_d = c'_0c_1\ldots c'_{k'-1}$ are* isomorphic, *denoted $S_d \sim S'_d$, iff $k = k'$ and there is a permutation τ of $\mathbf{d}$ such that for each $i, 0 \leq i < k$, $c'_i = \tau(c_i)$.*

2. *For any string x over $\mathbf{d}^*$, for each $i \in \mathbf{d}$, $I_x(i)$ is the 0-based index of the first occurrence (starting from the left) of i in x. ($I_x(i)$, by definition, is ∞ if i does not appear in x.)*

3. *A string x over $\mathbf{d}^*$ is* normalized *iff for each $i, j \in \mathbf{d}$, if at least one of i or j appear in x, then $[(I_x(i) < I_x(j))$ iff $(i < j)]$.*

4. *A d-snake $S_d = c_0 c_1 \ldots c_{(k-1)}$ is a* reversible *snake iff there exists a permutation σ of $\mathbf{d}$ such that for each $i, 0 \leq i < k$, $\sigma(c_i) = \sigma(c_{(k-1-i)})$. Let R_d^σ denote the set of reversible, maximal d-snakes as witnessed by σ.*

The following lemma is immediate from the definitions of isomorphic and normalized snakes.

Lemma 3 *For each snake S'_d there is a unique, normalized d-snake S_d such that $S_d \sim S'_d$.* □

Let $\mathcal{N}_d$ denote the set of all normalized, reversible, maximal d-snakes. Our aim here is to show an efficient algorithm for generating a set equivalent to $\mathcal{N}_d$.

Definition 3 *Suppose σ is a permutation of order 2. σ is a* normalized permutation *iff it has the form $(0) \ldots (k-1)(k, k+1), \ldots, (k + 2 \cdot ((d-k)/2), k + 2 \cdot ((d-k)/2) + 1)$, for some $k \geq 0$ such that $(d-k)$ is even. Such a normalized permutation has the signature $1^{\lambda_1} 2^{\lambda_2}$, where $\lambda_1 = k$ and $\lambda_2 = \frac{(d-k)}{2}$. For such a normalized permutation σ of $\mathbf{d}$ we will define the order of $i \in \sigma$, denoted by o_σ, as*

1. *for each $i < \lambda_1$, $o_\sigma(i) = 1$, and*

2. *for each $\lambda_1 \leq i < d$, $o_\sigma(i) = 2$.*

Let Σ_d be the set of all normalized permutations of $\mathbf{d}$.

Note that, for simplicity, we have defined the notion of a normalized permutation only for permutations of order 2; clearly, the notion can be extended to arbitrary permutations.

For any string x over $\mathbf{d}^*$, we will denote the reversal of the string by $\overline{x}$. Suppose S_d is a reversible snake as witnessed by a permutation σ of $\mathbf{d}$. Then, from the definition of reversible snakes, it follows that there exist strings x_0, x_1 over $\mathbf{d}^*$ where $|x_0| \leq 1$, so that $S_d = \sigma(\overline{x_1}) x_0 x_1$. We will use this fact in our generation process. Let `null` denote the empty string.

Now consider the Algorithm `Generate`.

Algorithm 1 `Generate` *(d)*

```
S = ∅;
for each σ ∈ Σ_d with signature 1^λ1 2^λ2
    if λ1 > 0 then
        HelpGenerate ("0", ∅, d, σ, S);
    HelpGenerate (null, ∅, d, σ, S);
return S;

end
```

For each $\sigma \in \Sigma_d$, the procedure `Generate` calls the procedure `HelpGenerate` first with the first parameter being the string containing 0 to generate all reversible (as witnessed by σ), maximal snakes of odd length, and then with the first parameter being the empty string to generate all reversible (as witnessed by σ), maximal snakes of even length. The parameter S, in `HelpGenerate` is used to accumulate the generated snakes.

Algorithm 2 `HelpGenerate` *(x_0, x_1, d, σ, S)*

if $\sigma(\overline{x_1})x_1$ is maximal **then**
    $S = S \cup \{\sigma(\overline{x_1})x_0x_1\}$;
else
    for each $i \in \mathbf{d}$
        if $((o_\sigma(i) = 1)$ **and** (every $j < i$ is in x_1))
        or $((o_\sigma(i) = 2)$ **and** (every $j < i$ such that $o_\sigma(j) = 2$ is in x_1 or in $\sigma(\overline{x_1})$)) **then**
            if $\sigma(i)\sigma(\overline{x_1})x_0x_1i$ is a d-snake **then**
                `HelpGenerate` (x_0, x_1i, d, σ, S);

end

Let the set of d-snakes generated by `Generate`(d) be denoted by $G(d)$. We will show that

Theorem 3 *For each d, $\sim$ is a bijection between $\mathcal{N}_d$ and $G(d)$.*

That is, it is adequate to consider only the permutations in Σ_d to generate all (up to isomorphism) reversible, maximal d-snakes, and it is easy to verify that Σ_d contains only $\lfloor \frac{d}{2} \rfloor$ permutations.

First, from the definition of a reversible snake, it can be easily verified that

Lemma 4 *Suppose S_d is a reversible d-snake as witnessed by the permutation σ. Then, σ is a permutation of order 2.* □

The following lemma follows from the definition of normalized permutations.

Lemma 5 *For each order 2 permutation σ of* $\mathbf{d}$*, there is a unique normalized permutation, denoted σ^N, with signature $1^{\lambda_1}2^{\lambda_2}$ such that*

$$|\{i \in \mathbf{d} | \sigma(i) = i\}| = \lambda_1.$$

□

Hence we define the following notion of equivalence of permutations, and then Lemma 6 gives a characterization of equivalent permutations.

Definition 4 *Two permutations σ_1 and σ_2 of* $\mathbf{d}$ *are* equivalent, *denoted $\sigma_1 \sim \sigma_2$ iff $(\exists\tau)[\sigma_1 = \tau \circ \sigma_2 \circ \tau^{-1}]$.*

Lemma 6 *Suppose* σ_1 *and* σ_2 *are permutations of* $\mathbf{d}$*. Then,*

$$(\sigma_1 \sim \sigma_2) \Longleftrightarrow \textit{ they both have the same signature.}$$

$\square$

Then we have the following two lemmas relating isomorphic reversible snakes and the equivalence of the permutations that witness the reversibility of these snakes.

Lemma 7 *Suppose* σ_1 *and* σ_2 *are permutations of* $\mathbf{d}$*. Suppose* S_d^1 *is a reversible, maximal d-snake as witnessed by* σ_1 *and* S_d^2 *is a reversible d-snake as witnessed by* σ_2*. Then,*

$$(S_d^1 \sim S_d^2) \Longrightarrow (\sigma_1 \sim \sigma_2).$$

PROOF. Since $S_d^1 \sim S_d^2$, there is a permutation τ of $\mathbf{d}$ such that $S_d^1 = \tau(S_d^2)$. Then, since S_d^1 is reversible as witnessed by σ_1 and S_d^2 is reversible as witnessed by σ_2,

$$\begin{aligned}
\sigma_1(S_d^1) &= \overline{S_d^1} \\
&= \overline{\tau(S_d^2)} \\
&= \tau(\overline{S_d^2}) \\
&= \tau \circ \sigma_2(S_d^2) \\
&= \tau \circ \sigma_2 \circ \tau^{-1}(S_d^1).
\end{aligned}$$

Since S_d^1 is a maximal snake, by Lemma 1, each element of $\mathbf{d}$ appears in S_d^1, and thus, $\sigma_1 \sim \sigma_2$. $\square$

Lemma 8 *Suppose* σ_1 *and* σ_2 *are permutations of* $\mathbf{d}$*. If* $\sigma_1 \sim \sigma_2$ *then for each d-snake* S_d^1 *that is reversible as witnessed by* σ_1*, there is a d-snake* S_d^2 *that is reversible as witnessed by* σ_2 *and such that* $S_d^1 \sim S_d^2$*.*

PROOF. Since $\sigma_1 \sim \sigma_2$, by Lemma 6, there is a τ such that $\sigma_1 = \tau \circ \sigma_2 \circ \tau^{-1}$. Also, since S_d^1 is reversible as witnessed by σ_1, $\sigma_1(S_d^1) = \overline{S_d^1}$. Therefore,

$$\tau \circ \sigma_2 \circ \tau^{-1}(S_d^1) = \overline{S_d^1}.$$

That is,

$$\sigma_2(\tau^{-1}(S_d^1)) = \tau^{-1}(\overline{S_d^1}) = \overline{\tau^{-1}(S_d^1)}.$$

Thus $\tau^{-1}(S_d^1)$ is a reversible snake as witnessed by σ_2, and by definition, $\tau^{-1}(S_d^1) \sim S_d^1$. $\square$

Definition 5 *Suppose* $\sigma \in \Sigma_d$*. Suppose* $S_d = \sigma(\overline{x_1})x_0x_1$ *is a reversible, maximal snake.* S_d *is* σ-normalized *iff*

1. $x_0 = 0$ *if* $|S_d|$ *is odd, and* $x_0 = \emptyset$ *otherwise, and*
2. *for each* $i, j \in \mathbf{d}$, $I_{x_0x_1}(i) < I_{x_0x_1}(j) \Longrightarrow [(i < j) \vee (o_\sigma(i) < o_\sigma(j))]$.

Let M_d^σ denote the set of σ-normalized d-snakes.

Lemma 9 *For each $\sigma \in \Sigma_d$, if $x \neq y \in M_d^\sigma$, then $x \not\sim y$.*

PROOF. Suppose, by way of contradiction, $x = \sigma(\overline{x_1})x_0x_1 \in M_d^\sigma$, $y = \sigma(\overline{y_1})y_0y_1 \in M_d^\sigma$, and $x \sim y$ via τ, where τ is not the identity. By Lemma 1, each $i \in \mathbf{d}$ occurs in x and in y. Then,

$$\forall i \in \mathbf{d}, o_\sigma(i) = o_\sigma(\tau(i)). \tag{4}$$

Therefore, $\forall i \in \mathbf{d}, I_{x_0x_1}(i) < I_{x_0x_1}(\tau(i)) \implies i < \tau(i)$. Suppose for each $i \in x_1$, $I_{x_0x_1}(i) \geq I_{x_0x_1}(\tau(i))$. Then, either τ is the identity giving us a contradiction, or for each $i \in x_1$, $i \neq \tau(i)$ and $\tau(i) \in x_1$ and so, $I_{x_0x_1}(\tau(i)) \geq I_{x_0x_1}(\tau^2(i))$. By Equation (4), $\tau^2(i) = i$, whenever $i \neq \tau(i)$. Thus, $I_{x_0x_1}(\tau^2(i)) \geq I_{x_0x_1}(i)$ contradicting our supposition just above. Therefore, there is an $i \in x_1$ such that $I_{x_0x_1}(i) < I_{x_0x_1}(\tau(i))$.

Let $i_0 \in \mathbf{d}$ so that $I_{x_0x_1}(i_0) = min\{I_{x_0x_1}(j) | j \neq \tau(j)\}$. Then, $I_{y_0y_1}(\tau(i_0)) = min\{I_{y_0y_1}(j) | j \neq \tau(j)\}$.

So we get, $I_{x_0x_1}(\tau(i_0)) > I_{x_0x_1}(i_0)$, implies $\tau(i_0) > i_0$, and $I_{y_0y_1}(\tau(i_0)) < I_{y_0y_1}(i_0)$, implies $\tau(i_0) < i_0$, giving us a contradiction. □

By the construction and the exhaustive nature of algorithm `Generate`, we have

Lemma 10

$$\bigcup_{\sigma \in \Sigma_d} M_d^\sigma = G(d).$$

□

Lemma 11 *For each reversible, maximal d-snake S_d, there exists a unique $\sigma \in \Sigma_d$ and a unique σ-normalized d-snake S'_d such that $S_d \sim S'_d$.*

PROOF. Suppose S_d is a reversible, maximal d-snake as witnessed by a permutation σ_1 of $\mathbf{d}$. Then $S_d = \sigma_1(\overline{x_1})x_0x_1$. Let $x_0 = a_0$ and $x_1 = a_1a_2 \ldots a_k$. By Lemma 5, there exists a unique $\sigma_1^N \in \Sigma_d$ such that $\sigma_1 \sim \sigma_1^N$. We will construct a permutation τ of $\mathbf{d}$ by running the following algorithm in stages 0 through k.

Algorithm 3 *Stage i*

if ($a_i \neq \emptyset$) and ($\tau(a_i)$ is as yet undefined) then
  let j be the least number, if any, in $\mathbf{d}$ such that
  1. $o_{\sigma_1}(a_i) = o_{\sigma_1^N}(j)$, and
  2. j is not yet in the range of τ.
  if such a j exists, then $\tau(a_i) = j$.
  if ($o_{\sigma_1}(a_i) = 2$) then
    $\tau(\sigma_1(a_i)) = \sigma_1^N(j)$.

end

There are two potential cases where the above algorithm might fail, and we will show that neither of the two cases ever arise.

Case 1 An element already in the domain of τ is reassigned.
This can happen only when a_i is not in the domain of τ at the beginning of this stage. Since σ_1^N is an order 2 permutation, this implies that $\sigma_1^N(j)$ is not yet in the domain of τ. Thus nothing is reassigned at this stage, and so this case can never arise.

Case 2 No such j exists in some stage.
Since S_d is a maximal snake, by Lemma 1, each element of $\mathbf{d}$ is either in x_0, x_1 or $\sigma_1(\overline{x_1})$. By construction, in each stage the algorithm adds, if at all, to the range of τ only those elements that are not already in the range of τ. By Case 1 above, no element in the domain of τ is reassigned a value. Thus at the beginning of each stage τ is a bijection. Then, since a_i is not empty and not in the domain of τ, there is at least one $j \in \mathbf{d}$ that is not yet in the range of τ. Thus this case will never arise.

Clearly, then, $\tau(S_d)$ is a σ_1^N-normalized d-snake, and, by definition, $\tau(S_d) \sim S_d$. □

Proof Theorem 3.
We need to show that for each snake $S_d \in \mathcal{N}_d$, there is a unique snake $g \in G(d)$ such that $S_d \sim g$. Lemmas 11 and 10 assert the existence of such a g and Lemma 9 asserts the uniqueness of g. □

Thus we have shown that it is adequate to consider only the normalized permutations in generating all unique (up to isomorphism) reversible, maximal d-snakes. That is, we can restrict ourselves to *only* $\lfloor(d/2)\rfloor$ permutations of $\mathbf{d}$.

References

[AK88] H. L. Abbott and M. Katchalski. On the snake-in-the-box problem. *Journal of Combinatorial Theory*, 45:13–24, 1988.

[AK91] H. L. Abbott and M. Katchalski. On the construction of snake in the box codes. *Utilitas Mathematica*, 40:97–116, 1991.

[Bla64] W. L. Black. Electronic combination locks. Technical Report 73, Research Laboratory of Electronics, M. I. T., 1964.

[CFT64] R. T. Chien, C. V. Freiman, and D. T. Tang. Error correction and circuits on the n-cube. In *Proceedings of the 2nd Annual Allerton Conference on Circuit and System Theory*, pages 899–912, 1964.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability.* Freeman, 1979.

[HHW88] F. Harary, J. P. Hayes, and H. J. Wu. A survey of the theory of hypercube graphs. *Computer Math. Applications*, 15:277–289, 1988.

[Jab74] S. V. Jablonskiĭ. Discrete mathematics and mathematical problems of cybernetics. *Nauka*, 1, 1974.

[Kau58] W. H. Kautz. Unit distance error checking codes. *IRE Transactions on Electronic Computers*, 7:179–180, 1958.

[Kle70] V. Klee. The use of circuit codes in analog to digital conversion. In B. Harris, editor, *Graph Theory and its Applications*. Academic Press, New York, 1970.

[PT96] Kenneth G. Paterson and Jonathan Tuliani. Some new circuit codes, 1996.

[Sne94] H. S. Snevily. The snake-in-the-box problem: a new upper bound. *Discrete Mathematics*, 133(1-3):307–314, October 1994.