Research Article

An Algorithm for Generating a Family of Alternating Knots

Carlos Velarde, Ernesto Bribiesca, and Wendy Aguilar

Departamento de Ciencias de la Computación, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Apartado. 20-726, 01000 México City, DF, Mexico

Correspondence should be addressed to Ernesto Bribiesca, ernesto@leibniz.iimas.unam.mx

Received 25 September 2011; Accepted 11 October 2011

Academic Editors: C. Munuera and A. Vourdas

Copyright © 2012 Carlos Velarde et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An algorithm for generating a family of alternating knots (which are described by means of a chain code) is presented. The family of alternating knots is represented on the cubic lattice, that is, each alternating knot is composed of constant orthogonal straight-line segments and is described by means of a chain code. This chain code is represented by a numerical string of finite length over a finite alphabet, allowing the usage of formal-language techniques for alternating-knot representation. When an alternating knot is described by a chain, it is possible to obtain its mirroring image in an easy way. Also, we have a compression efficiency for representing alternating knots, because chain codes preserve information and allow a considerable data reduction.

1. Introduction

In 1993, Livingston [1] stated that “knot theory remains a lively topic today. Many of the basic questions, some dating to Tait’s first paper in the subject, remain open. At the other extreme, the results of recent years promise to provide many new insights”. Knot theory is a branch of algebraic topology. The three main techniques of knot theory are: geometric techniques, combinatorial methods, and algebraic tools. A knot K is a simple closed polygonal curve in three-dimensional Euclidean space $\mathbb{R}^3$ [1]. Knots are cataloged in order of increasing complexity. One measure of complexity that is often used is the crossing number, that is, the number of double points in the simplest planar projection of the knot. There is only one knot with crossing number of three (ignoring mirror reflections) the trefoil. The figure-eight knot is the only knot with a crossing number of four. There are two knots with a crossing number of five, three knots with a crossing number of six, and seven knots with a crossing number of seven. From there on the numbers increase dramatically. There are 12,965 knots with a crossing number of 13 or fewer crossings in a minimal projection. A complex knot is a knot

Figure 1: An example of a complex knot: its continuous and discrete version, respectively.

with a huge crossing numbers. Imagine, how many knots are there in an alternating knot which is represented by a complex knot with a crossing number of 1200? This paper deals with complex knots of this order of crossing numbers. Z^3 is a more computer-friendly place for generating knots. In $\mathcal{R}^3$, every point in space has coordinates drawn from the set of real numbers. Thus, the generation of complex knots in $\mathcal{R}^3$ produces a very complex computation. On the other hand, in Z^3 all points have integer coordinates. Thus, we represent complex knots which are embedded on the cubic lattice. Complex knots are composed of constant orthogonal straight-line segments and are represented by means of a chain code [2] which produces a numerical string of finite length over a finite alphabet, allowing the usage of grammatical techniques for alternating-knot representation. The above mentioned generates a very simple computation.

Several authors have analyzed knots on the cubic lattice. In 1993, Diao presented the minimal knotted polygons on the cubic lattice [3] and, in 1994, the number of smallest knots on the cubic lattice [4]. van Rensburg and Promislow define the minimal knots in the cubic lattice [5]. Hayes [6] presents the “Square Knots” on the cubic lattice, which are simple closed polygonal curves embedded in Z^3 . Other interesting theories: Kauffman [7] introduces the “Virtual Knot Theory”, which includes general Gauss codes. Nakamura and Rosenfeld [8] define the “Digital Knots” which represent an initial effort at the study of the concepts of knottedness and linkedness for digital objects. Complex knots are embedded on the cubic lattice. Thus, these knots are composed of constant orthogonal straight-line segments and are represented by chain coding [2]. In order to have a self-contained paper, Appendix A describes the used chain code. The left-hand side of Figure 1 shows an example of a complex knot in a continuous representation. The right-hand side of Figure 1 illustrates the same knot embedded on the cubic lattice. In the content of this work, knots are represented as ropes. This improves the understanding of the figures.

This paper is organized as follows: Section 2 describes the proposed family of rectangular alternating links. In Section 3, we present some results. Section 4 gives some conclusions. Finally, we present Appendix A which describes the used chain code and Appendix B which presents the program of the proposed algorithm.

2. A Family of Rectangular Alternating Links

The alternating link diagrams of the type illustrated in Figure 2 are used here to construct links like that shown in Figure 1. Each diagram is a set of rectilinear polygons and can be characterized in terms of two positive integers a and b . In what follows we present such a

Figure 2: An alternating link diagram composed of a set of rectilinear polygons with vertices on the 2-dimensional integer lattice. The polygons are bounded by an axis-aligned square, S_a , which has side length a and has the origin at lower left corner. They are also inscribed in a rectangle, $R_{a,b}$, which has its sides parallel to the diagonals of S_a , is symmetric with respect to the central point of S_a , and has sides through the points $(0, a - b)$ and $(0, a - b + 1)$. The vertices of the polygons are the lattice nodes lying on the sides of $R_{a,b}$ and they are numbered from 0 to $2a + 1$.

characterization, from which we obtain some properties of the family of polygons, useful to construct an algorithm for generating a chain code representation of the corresponding link.

Let a and b be integers such that $0 < b \leq a$. Let S_a be a square on the 2-dimensional integer lattice, with one corner at the origin and another at the point (a, a) . Let $R_{a,b}$ be the rectangle with sides parallel to the diagonals of S_a , symmetric with respect to the central point of S_a , and with sides through the points $(0, a - b)$ and $(0, a - b + 1)$.

The $2(a + 1)$ grid nodes lying on the edges of $R_{a,b}$ serve as vertices of a set $\mathcal{P}_{a,b}$ of rectilinear polygons, each of whose sides is the vertical or horizontal line segment that joins a corresponding pair of nodes. In Figure 2 those vertices appear counterclockwise enumerated from 0 to $2a + 1$, beginning with the node at coordinates $(0, a - b)$. Accordingly, let $V_a = \{0, 1, \dots, 2a + 1\}$ be the set of vertices of $\mathcal{P}_{a,b}$.

Let $v : V_a \rightarrow V_a$ be the function such that $v(i)$ is the vertex of $\mathcal{P}_{a,b}$ vertically opposed to vertex i . As a consequence of the way the enumeration of the vertices was done, we have that $i + v(i) = 2a + 1$. Then, from this equation and defining $n = 2(a + 1)$, we arrive to

$$v(i) = n - 1 - i. \quad (2.1)$$

Now, let $h : V_a \rightarrow V_a$ be the function such that $h(i)$ is the vertex horizontally opposed to vertex i . There are two cases: $i + h(i) = 2a - 2b + 1 = n - 2b - 1$, if $i \leq 2a - 2b + 1$, and $i + h(i) = 4a - 2b + 3 = 2n - 2b - 1$, otherwise. Taking into account that $h(i) < n$, in both cases we obtain

$$h(i) = (-2b - 1 - i) \bmod n. \quad (2.2)$$

Remark 2.1. (1) The functions v and h are bijections, they are self-inverse, and each one maps odd numbers to even numbers and vice versa.

(2) By the above remark, also the function composition $v \circ h : V_a \rightarrow V_a$ is bijective, but preserves the parity of its argument.

(3) The functions v , h , and $v \circ h$, being bijections, are elements of the group S_{V_a} of symmetries of V_a . In particular, we call attention to $\langle v \circ h \rangle$ —the cyclic subgroup generated by $v \circ h$ on S_{V_a} —and to its (induced) action over V_a .

(4) Along any polygon $P \in \mathcal{P}_{a,b}$, the horizontal sides alternate with the vertical ones. Thus, the consecutive vertices of P can be obtained by iterating alternate applications of h and v . For example, with $a = 9$ and $b = 4$ (which is the particular case presented in Figure 2), for the polygon through vertex 0, the corresponding iteration produces the (periodic) sequence

$$0, h(0), v(h(0)), h(v(h(0))), \dots \quad (2.3)$$

which has

$$0, 11, 8, 3, 16, 15, 4, 7, 12, 19 \quad (2.4)$$

as a shortest period.

(5) From the first remark, it follows that in sequences like those just presented the elements alternate in parity. So, if U is the set of even (odd) vertices of P , then $h(U)$ and $v(U)$ both are equal to the set of odd (even) vertices of P .

Now, to count the number of polygons in $\mathcal{P}_{a,b}$, it is useful to interpret this set as a graph, in the obvious way, and then proceed to count the number of cycles in this graph.

Definition 2.2. Let a , b , V_a , v , and h , be as above. $G_{a,b} = (V_a, E_{a,b})$ is the graph with vertex-set V_a and edge-set $E_{a,b} = \{\{i, v(i)\} \mid i \in V_a\} \cup \{\{i, h(i)\} \mid i \in V_a\}$.

We can see Figure 2 also as a drawing of $G_{a,b}$. Since each vertex is an end point of exactly two edges, $G_{a,b}$ is a regular graph of degree 2. Therefore, every component of $G_{a,b}$ is a cycle [9]. Now we will see how many components have G and how long they are.

Lemma 2.3. *The graph $G_{a,b}$ defined above is composed of $c = \gcd(a+1, b)$ cycles of length n/c , and all the vertices $0, 1, \dots, c-1$ belong to distinct cycles.*

Proof. From (2.1) and (2.2) we have that $v(h(i)) = n-1 - (-2b-1-i) \bmod n = (2b+i) \bmod n$, that is

$$(v \circ h)(i) = (2b+i) \bmod n. \quad (2.5)$$

Now, to analyze the action of $\langle v \circ h \rangle$ over V_a (see Remark 2.1 (3)), it is useful the embedding of V_a into the additive group $\mathbb{Z}/n$. From (2.5) it follows that $(v \circ h)^q(0) = q(2b) \bmod n$, for $q \in \mathbb{N}$. Hence, the orbit of 0 under the action of $\langle v \circ h \rangle$ is the (finite) set $\{q(2b) \bmod n \mid q = 0, 1, 2, \dots\}$ which is the (cyclic) subgroup of $\mathbb{Z}/n$ generated by $2b$. Then, this subgroup has $c' = \gcd(n, 2b)$ as its least generator, it is of order n/c' [10, Section 6.5-6], and the set $\{[i] \mid 0 \leq i < c'\}$ of its c' cosets coincide with the set of orbits of $\langle v \circ h \rangle$ on $\mathbb{Z}/n$. But the set of vertices of a cycle of $G_{a,b}$ is equal to the union of two of these orbits (see Remark 2.1(5)). Hence, there are $c = c'/2 = \gcd(n, 2b)/2 = \gcd(a+1, b)$ cycles in $G_{a,b}$, each one of length $2(n/c') = n/c$. Now,

Figure 3: (a) A link for the alternating link diagram shown in Figure 2, (b) a section of the link, and (c) the pieces of a section.

in terms of cosets, the set of vertices of a cycle of $G_{a,b}$ is equal to a union of the form $[i] \cup v([i])$, with $0 \leq i < c'$. By (2.1), and because $n-1-i$ and $c'-1-i$ are in the same coset, we obtain that

$$v([i]) = [v(i)] = [n-1-i] = [c'-1-i]. \quad (2.6)$$

Therefore, the c cycles have, respectively, the following sets of vertices:

$$[i] \cup [(c'-1)-i], \quad \text{for } 0 \leq i < \frac{c'}{2} = c. \quad (2.7)$$

□

Next, we will define a link $\mathcal{L}_{a,b}$ for the alternating link diagram $\mathcal{P}_{a,b}$. Figure 3(a) shows $\mathcal{L}_{9,4}$. To begin with, we state some additional observations on Figure 2.

Remark 2.4. (1) There are b and $a+1-b$ vertices of $\mathcal{P}_{a,b}$ in the upper left side and in the lower left side of $R_{a,b}$, respectively. Let

$$b' = \min(b, a+1-b). \quad (2.8)$$

Since $c = \gcd(a+1, b) = \gcd(a+1, b') \leq b'$, the vertices $0, 1, \dots, c-1$ lie on the lower left side of $R_{a,b}$.

(2) Let e be an edge of a polygon $P \in \mathcal{P}_{a,b}$. If the end points of e lie on parallel sides of $R_{a,b}$, then e is of even length $2b'$, otherwise e has odd length. Furthermore, e has odd length if and only if e is at distance $d < b'$ from a side of S_a parallel to e , and in this case e is of length $2d+1$. Thus, the lengths of the vertical and horizontal edges through the vertex i are, respectively, given by the functions $l_v, l_h : V_a \rightarrow \mathbb{N}$ such that

$$l_v(i) = \begin{cases} 2j+1, & \text{if } j < b', \\ 2(a-j)+1, & \text{if } j > a-b', \\ 2b', & \text{otherwise,} \end{cases} \quad \text{with } j = i \bmod (a+1), \quad (2.9)$$

$$l_h(i) = l_v(i+b). \quad (2.10)$$

For example, the alternate application of l_h and l_v to the vertices obtained for the polygon P in Remark 2.1(4) gives the following sequence of edge lengths:

$$8, 3, 5, 7, 1, 8, 3, 5, 7, 1. \quad (2.11)$$

(3) If e has even length, then its adjacent edges lie on distinct semiplanes with respect to the line containing e , otherwise they are to the same side of such a line.

The link $\mathcal{L}_{a,b}$ is constructed in such a way that $\mathcal{P}_{a,b}$ is the projection of $\mathcal{L}_{a,b}$ on the plane $x = 0$, under the mapping $(x, y, z) \mapsto (0, y, z)$. Let $K_i \in \mathcal{L}_{a,b}$ be the component knot whose projection is the polygon $P_i \in \mathcal{P}_{a,b}$ through vertex $i \in \{0, 1, \dots, c-1\}$, with c as defined in the preceding lemma. We orient P_i by departing from i to the right, that is, following the cycle $i, h(i), \dots, i$. The knot K_i is an axis-aligned polygon whose vertices lie on the planes $x = 0$ and $x = 1/2$, with i as the starting vertex. In turn, this polygon is a concatenation of “sections,” which are planar polygonal curves corresponding to the edges of P_i , similar to that shown in Figure 3(b).

The section s related to a directed edge e is a concatenation of $|e|$ “pieces” like the three shown in Figure 3(c); s begins with a horizontal piece, which belongs either to the plane $x = 0$ or to the plane $x = 1/2$ and is followed by $|e| - 1$ alternate occurrences of the other two pieces. If we split the first piece at its midpoint, s can be seen as a directed polygonal composed of line segments of constant length $(1/2)$, as delineated in Figure 4(a) which also shows the code obtained by means of (A.1) for each vertex apart from the first and fourth ones, whose codes p and q remain unknown until we have the placement of s along the knot (but $q \neq 0$ since q labels a right angle). Thus, in formal language terminology, a section is a chain over $\Sigma = \{0, 1, 2, 3, 4\}$ of the form either $p0$ or $p00q2(042)^*$, where $p, q \in \Sigma$ and $q \neq 0$, with the interpretation given in A. Figures 4(b)–4(f) show some configurations that correspond to five of the 20 possible values for the pair (p, q) . Next we will prove that only the first two illustrated cases occur in K_i .

Lemma 2.5. *Let e be an edge of P_i and let s be the section of K_i corresponding to e . Then, s has $p = 3$ or 4 , and, if $|e| > 1$, $q = 1$.*

Proof. The proof goes by induction on the position of s along K_i . If s is the first section of K_i , then its initial point is vertex i which, by Remark 2.4(2.1), lies on the lower left side of $R_{a,b}$. Thus, the right angle previous to i along K_i must belong to one of the two configurations shown in Figures 4(b)–4(c), depending on the length d' of the edge of P_i previous to e ; if $d' = 1$, then $p = 4$, and if $d' > 1$, then $p = 3$. Furthermore, if $|e| > 1$ (thus there is a point for q in s), $q = 1$ in both cases.

Now, suppose by induction that a section s satisfies the lemma. We want to show that the section s'' next to s also satisfies the lemma. Let s' be the section previous to s . The induction hypothesis implies that the configuration for s' and s must be one of those in Figures 4(b)–4(c), up to a rotation. Suppose $|e|$ is odd; then, as a consequence of Remark 2.4(3), s' and s'' are to the same side of the plane containing s and, since s jumps $|e| - 1$ times between the planes $x = 0$ and $x = 1/2$, the end points of s lie both in the same of these planes; therefore, the configuration determined by s and s'' must be also, up to a rotation, one of those shown in Figures 4(b)–4(c). On the other hand, if $|e|$ is even, from the same remark it follows that s' and s'' are in opposite sides of the plane of s and, since now $|e| - 1$ is odd, the end points of s lie one in the plane $x = 0$ and the other in the plane $x = 1/2$, facts that lead to the same conclusion obtained in the first case. $\square$

Figure 4: (a) Chain code for a section of a link. To calculate p and q , a corresponding right angle of reference is needed; the configurations (b) to (f) illustrate the cases $(p, q) = (4, 1), (3, 1), (4, 3), (1, 3)$; and $(2, 3)$.

Corollary 2.6. Let e' and e be two successive edges of P_i , of lengths d' and d , respectively. Then the chain code of the section s in K_i corresponding to e is given by the following function:

$$\kappa(d', d) = \begin{cases} 40 & \text{if } d' = 1, d = 1, \\ 30 & \text{if } d' > 1, d = 1, \\ 40 \ 012(042)^{d-2} & \text{if } d' = 1, d > 1, \\ 30 \ 012(042)^{d-2} & \text{if } d' > 1, d > 1. \end{cases} \quad (2.12)$$

The following program, written in *Haskell* language [11, 12], is a computable version of the function κ :

$$\begin{aligned} \text{section } (d', d) = & \text{concat } ((\text{if } d' == 1 \text{ then "40" else "30"}) \\ & : \text{take } (d - 1) \text{ ("012" : repeat "042"})), \end{aligned} \quad (2.13)$$

For example, `section(1,8)` evaluates to "40012042042042042042". The next function calculates the chain code for K_i :

$$\text{chain } i = \text{concat}[\text{section}(d', d) \mid (d', d) \leftarrow \text{side_length_pairs } i], \quad (2.14)$$

where `side_length_pairs i` gives the list of pairs (d', d) needed to compute the successive sections of K_i . The definitions of these functions, as well as those for v , h , l_v , l_h , n , c , and b' , are local to the main function, which computes the list of strings of code for the c cycles of the link $\mathcal{L}_{a,b}$:

$$\text{carpet } a \ b = [\text{chain } i \mid i \leftarrow [0 \dots c - 1]], \quad (2.15)$$

The complete definition of this program is presented in Appendix B.

3. Results

In order to probe our proposed method, we present some examples of alternating-knot generation. Figure 5 illustrates an example of an alternating knot of 22×12 ($a = 22, b = 12$).

The chain of the alternating knot shown in Figure 5 is as follows:

[illegible]

If we replace the chain elements 1 by the chain elements 3 and vice versa in the above-mentioned chain, we obtain the mirroring image [2] of the alternating knot illustrated in Figure 5.

Figure 6 illustrates another example of an alternating knot of 50×25 using the proposed method.

4. Conclusions

We have presented a modest attempt for generating alternating knots which are represented by means of chain coding. The chain-code representation of alternating knots preserves information and allows a considerable data reduction. Also, the mirror images of alternating knots are obtained in an easy way.

Appendices

A. The Chain Code

In order to have a self-contained paper, we summarize the main concepts and definitions of the used chain code. In the content of this paper, we use this chain code to represent complex knots.

Definition A.1. A discrete knot K_D is the digitalized representation of a knot K and is composed of constant orthogonal straight-line segments, whose direction changes are described as a chain.

Figure 5: An example of an alternating knot of 22×12 .

Definition A.2. A *chain* A is a finite sequence of elements and is represented by $A = a_1 a_2 \cdots a_n = \{a_i : 1 \leq i \leq n\}$, where n indicates the length of the chain.

The chain elements for a discrete knot are obtained by calculating the relative orthogonal direction changes of the contiguous constant straight-line segments along the knot. There are only five possible orthogonal direction changes [13, 14] for representing any discrete knot. Thus, each discrete knot is represented by a chain of elements. Figure 7(b) illustrates an example of a discrete complex knot.

An element a_i of a chain, taken from the set $\{0, 1, 2, 3, 4\}$, labels a vertex of the discrete knot and indicates the orthogonal direction change of the polygonal path in such a vertex. Figure 7(a) summarizes the rules for labeling the vertices: to a straight-angle vertex, a 0 is attached; to a right-angle vertex corresponds one of the other labels, depending on the position of such an angle with respect to the preceding right angle in the polygonal path.

In order to improve the understanding of the chain elements, we have colored the straight-line segments which are defined by their corresponding chain elements. Thus, the straight-line segment defined by the chain element 0 in green, 1 in cyan, 2 in yellow, 3 in magenta, and 4 in red, respectively. This is valid for the web version of the paper; however, the gray-level version of the paper also allows us to understand the above-mentioned notation. Formally, if the consecutive sides of the reference angle have respective directions $\mathbf{b}$ and $\mathbf{c}$ (see Figure 7(a)), and the side from the vertex to be labeled has direction $\mathbf{d}$ (here, by direction we understand a unit vector), then the label or chain element is given by the following function, where $\times$ denotes the vector product in $\mathbb{R}^3$:

$$\text{chain_element}(\mathbf{b}, \mathbf{c}, \mathbf{d}) = \begin{cases} 0 & \text{if } \mathbf{d} = \mathbf{c}, \text{ (in green),} \\ 1 & \text{if } \mathbf{d} = \mathbf{b} \times \mathbf{c}, \text{ (in cyan),} \\ 2 & \text{if } \mathbf{d} = \mathbf{b}, \text{ (in yellow),} \\ 3 & \text{if } \mathbf{d} = -(\mathbf{b} \times \mathbf{c}), \text{ (in magenta),} \\ 4 & \text{if } \mathbf{d} = -\mathbf{b}, \text{ (in red).} \end{cases} \quad (\text{A.1})$$

Thus, the procedure to find the chain of a discrete knot is as follows.

- (i) *Select* an arbitrary vertex of the discrete knot as the origin. Also, *select* a direction for traveling around the discrete knot. Figure 7(b) illustrates the selected origin which is represented by a sphere. Also, the selected direction is to the right.
- (ii) *Compute* the chain elements of the discrete knot. Figure 7(b) shows the first element of the chain which corresponds to the element 0. The second element corresponds

Figure 6: An example of an alternating knot of 50×25 .

to the chain element 0, too. The third element corresponds to the chain element 1. Note that when we are traveling around a discrete knot, in order to obtain its chain elements and find zero elements, we need to know what nonzero element (labeling a right angle) was the last one in order to define the next element. Finally, we obtain the following chain:

$$00120420423040012042300120423040012042042304001204230012042304. \quad (\text{A.2})$$

The main characteristics of this chain code are as follows.

- (i) It is invariant under translation and rotation. This is due to the fact that only relative orthogonal direction changes are used.
- (ii) In this code, there are only five possible orthogonal direction changes for representing any discrete knot, this produces a numerical string of finite length over a finite alphabet, allowing the usage of grammatical techniques for complex-knot generation.
- (iii) Using this code, it is possible to obtain the *mirror image* of a discrete knot with ease. The chain of the mirror image of a discrete knot is another chain (termed the *reflected chain*) whose elements 1 are replaced by elements 3 and vice versa. This replacement does not depend on the orthogonal reflecting plane used, it is valid for the three possible orthogonal mirroring planes. We do not prove this, only we illustrate it [13].

A complete review of the above-mentioned code can be found in [2, 13].

B. The Program

Here is the function *carpet* that computes the links presented in Section 2. Let $a, b \in \mathbb{Z}$; if $0 < b \leq a$, (*carpet a b*) returns the list of chains of code for the knots of $\mathcal{L}_{a,b}$, otherwise the returned list is empty.

Figure 7: An example of a complex knot.

```

carpet a b = if 0 < b ≤ a then [chain i | i ← [0...c-1]] else []
where
  m = a + 1; n = 2 × m; c = gcd m b; b' = min b (m - b)
  v i = n - 1 - i—See (2.1)
  h i = (-2 × b - 1 - i) mod n—See (2.2)
  lv i | j < b' = 2 × j + 1—See (2.9)
        | j > a - b' = 2 × (a - j) + 1
        | otherwise = 2 × b'
  where j = i mode m
  lh i = lv (i + b)—See (2.10)
  cycle i = take (n div c) (iterate2 h v i)—See Remark 2.1(4)
  side_lengths i = map2 lh lv (cycle i)—See Remark 2.4(2)
  side_length_pairs i = zip (last ds : ds) ds
  where ds = side_lengths i
  section (d', d) = concat ((if d' == 1 then "40" else "30")—See (2.12)
    : take (d - 1) ("012" : repeat "042"))
  chain i = concat [section (d', d) | (d', d) ← side_length_pairs i]

```

The functions *iterate2* and *map2* have definitions similar to those of the standard functions *iterate2* and *map* provided by *Haskell*. The invocation *(iterate2 fg x)* returns the infinite list $[x, fx, g(fx), f(g(fx)), \dots]$ of repeated and alternate applications of *f* and *g* to *x*:

$$\begin{aligned} \text{iterate2} &:: (a \longrightarrow a) \longrightarrow (a \longrightarrow a) \longrightarrow a \longrightarrow [a] \\ \text{iterate2 } f \ g \ x &= x : \text{iterate2 } g \ f \ (f \ x) \end{aligned} \quad (\text{B.1})$$

and *(map2 fg [x₀, x₁, x₂, x₃, ...])* returns the list $[f \ x_0, g \ x_1, f \ x_2, g \ x_3, \dots]$ of alternate applications of *f* and *g* to the successive elements of the list $[x_0, x_1, x_2, x_3, \dots]$:

$$\begin{aligned} \text{map2} &:: (a \longrightarrow b) \longrightarrow (a \longrightarrow b) \longrightarrow [a] \longrightarrow [b] \\ \text{map2 } fg \ [] &= [] \\ \text{map2 } fg \ (x : xs) &= fx : \text{map2 } g \ f \ xs. \end{aligned} \quad (\text{B.2})$$

Acknowledgment

The authors wish to express their gratitude to Guillermo Rojas for his help in generating patterns of alternating knots.

References

- [1] C. Livingston, *Knot Theory*, vol. 24, Mathematical Association of America, Washington, DC, USA, 1993.
- [2] E. Bribiesca, "A method for computing families of discrete knots using knot numbers," *Journal of Knot Theory and Its Ramifications*, vol. 14, no. 4, pp. 405–424, 2005.
- [3] Y. Diao, "Minimal knotted polygons on the cubic lattice," *Journal of Knot Theory and Its Ramifications*, vol. 2, no. 4, pp. 413–425, 1993.
- [4] Y. Diao, "The number of smallest knots on the cubic lattice," *Journal of Statistical Physics*, vol. 74, no. 5–6, pp. 1247–1254, 1994.
- [5] E. J. J. van Rensburg and S. D. Promislow, "Minimal knots in the cubic lattice," *Journal of Knot Theory and Its Ramifications*, vol. 4, no. 1, pp. 115–130, 1995.
- [6] B. Hayes, "Square knots," *American Scientist*, vol. 85, no. 6, pp. 506–510, 1997.
- [7] L. H. Kauffman, "Virtual knot theory," *European Journal of Combinatorics*, vol. 20, no. 7, pp. 663–690, 1999.
- [8] A. Nakamura and A. Rosenfeld, "Digital knots," *Pattern Recognition*, vol. 33, no. 9, pp. 1541–1553, 2000.
- [9] F. Harary, *Graph Theory*, Addison-Wesley, 1969.
- [10] C. R. Jordan and D. A. Jordan, *Groups*, Arnold, London, UK, 1994.
- [11] R. Bird, *Introduction to Functional Programming Using Haskell*, Prentice Hall, 1998.
- [12] S. P. Jones, Ed., *Haskell 98 Language and Libraries. The Revised Report*, 2002, <http://www.haskell.org/>.
- [13] E. Bribiesca, "A chain code for representing 3D curves," *Pattern Recognition*, vol. 33, no. 5, pp. 755–765, 2000.
- [14] A. Guzmán, "Canonical shape description for 3-d stick bodies," Tech. Rep. ACA-254-87, MCC, Austin, Tex, USA, 1987.