In *The Braider* issue No. 5 we discussed *the complementary cyclic bight-number scheme* for Regular Knots and *the algorithm diagram* for Column-coded Regular Knots. In this issue we will look at them in assosiation with Semi-Regular Knots. Recall that Regular Knots are Regular Cylindrical Braids in which the number of parts and the number of bights are **coprime**, and that Semi-Regular Knots are Regular Cylindrical Braids in which the number of parts and the number of bights have a greatest common divisor greater than 1, hence are not coprime. Also recall that for a Regular Cylindrical Braid the greatest common divisor of its parts and bights is equal to the number of strings required in its construction. For easy reference we have in the diagrams placed the starting-points of the strings in adjacent bight-points, however, it should be needless to say that in the construction of the braid we distribute the starting-points of the strings as evenly as possible along a bight-boundary.

The distribution of the actual starting-points has no influence on the complementary cyclic bight-number scheme, nor has it any influence on the algorithm diagram for Column-coded Semi-Regular Knots. Only the string-sequence of braiding affects the complementary cyclic bight-number scheme and hence the algorithm diagram.

The string-run diagrams in Fig. 154 are of Semi-Regular Knots which have 20-parts and 15-bights. Since the greatest common divisor of 20 and 15 is equal to 5, we require five strings in their construction. The strings are numbered in the order in which they will be braided.

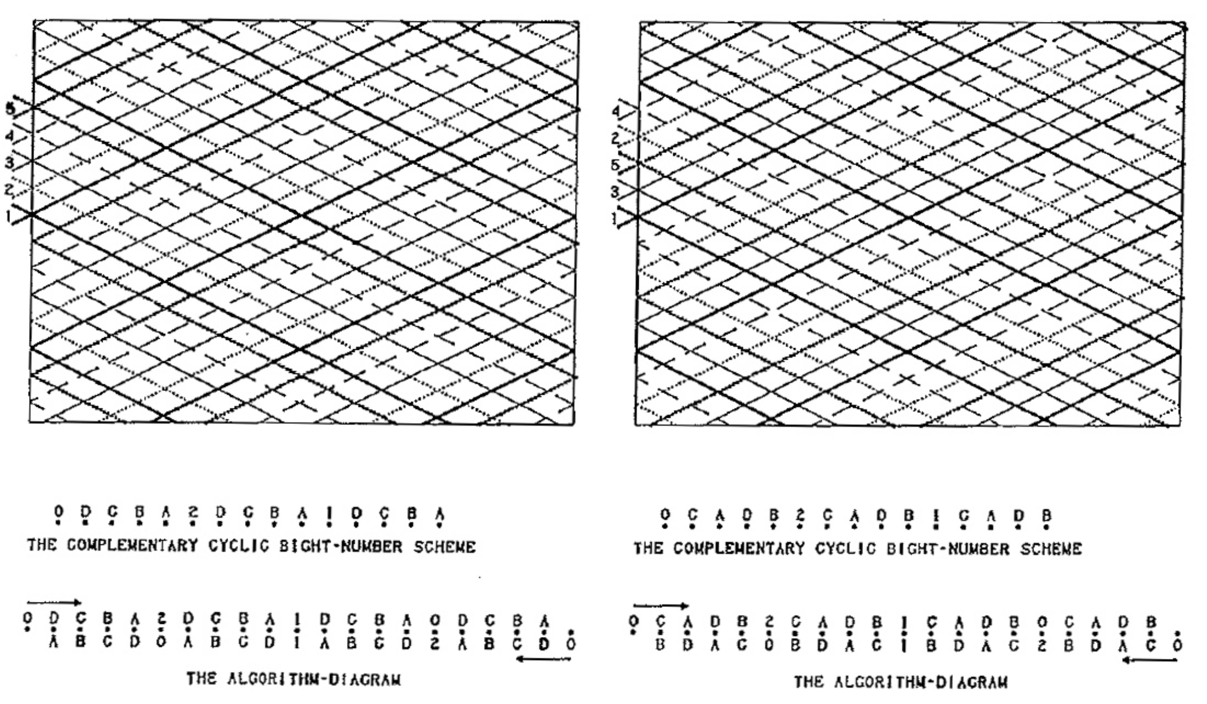
5
4
3
2
1
4
2
5
3
1
0 D C B A 2 D C B A 1 D C B A
THE COMPLEMENTARY CYCLIC BIGHT-NUMBER SCHEME
0 C A D B 2 C A D B 1 C A D B
THE COMPLEMENTARY CYCLIC BIGHT-NUMBER SCHEME
0 D C B A 2 D C B A 1 D C B A 0 D C B A
A B C D 0 A B C D 1 A B C D 2 A B C D 0
THE ALGORITHM-DIAGRAM
0 C A D B 2 C A D B 1 C A D B 0 C A D B
B D A C 0 B D A C 1 B D A C 2 B D A C 0
THE ALGORITHM-DIAGRAM

Fig. 154 — Two examples of regular periodic sequences for braiding the strings.

Since any string-sequence of braiding is cyclic, we obtain for n strings a total of $(n-1)!$ distinctly different sequences.[†] Let there be m natural numbers, each one smaller than n, which are coprime with n. Then the $(n-1)!$ distinctly different string-sequences of braiding the n strings contain m sequences which are regular periodic. For example, when $n = 9$ we have $(n-1)! = (9-1)! = 8! = 40320$ distictly different string-sequences, which contain $m = 6$ sequences which are regular periodic (the natural numbers $1, 2, 4, 5, 7, 8$ are coprime with 9). These six regular periodic string-sequences are:

$$1,2,3,4,5,6,7,8,9.$$
$$1,3,5,7,9,2,4,6,8.$$
$$1,5,9,4,8,3,7,2,6.$$
$$1,6,2,7,3,8,4,9,5.$$
$$1,8,6,4,2,9,7,5,3.$$
$$1,9,8,7,6,5,4,3,2.$$

Only for **regular** periodic string-sequences will a single complementary cyclic bight-number scheme suffice, and hence for column-coded Semi-Regular Knots a single algorithm diagram is all that is needed for braiding the knot.

For the diagrams in Fig. 154, the respective string-sequences are $1, 2, 3, 4, 5$ and $1, 3, 5, 2, 4$. These sequences are **regular** periodic. Since $p = 20$ and $b = 15$, we obtain $|-p|_b = |-20|_{15} = 10$. Consequently we obtain the following complementary cyclic bight-number scheme for each single string:

```
0         2         1
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Of the 15 bight-points only 3 have received a bight-number ($b \div$ (g.c.d. of p and b) $= 15 \div 5 = 3$).

For every string on its own we can use this scheme, and in doing so the '0' point does not stay in the same bight-point of the knot, but shifts to the bight-point of the string under consideration. Strings which have already been laid down occupy their associated bight-points relative to the bight-points of the string under consideration, hence relative to the numbered bight-points in our above complementary cyclic bight-number scheme. Thus for the string-run of the left-hand knot in Fig. 154 we obtain the following: First we lay down string #1; for this string the above complementary cyclic bight-number scheme applies. Then we lay down string #2, but since string #1 already occupies the bight-points immediately **below** those which string #2 is going to occupy, we mark the bight-points immediately to the **left** (in cyclic fashion of course) of those who possess numerical values with A. Hence for string #2 we obtain the complementary cyclic bight-number scheme:

```
0        A  2        A  1        A
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Next we lay down string #3, but since strings #1 and #2 already occupy the two consecutive bight-points immediately **below** those which string #3 is going to occupy, we mark the bight-points immediately to the **left** (in cyclic fashion of course) of those who possess the letter A with B. Hence for string #3 we obtain the complementary cyclic bight-number scheme:

```
0     B  A  2     B  A  1     B  A
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Although the 'A' points were occupied later than the 'B' points, this is irrelevant for our purpose since both the 'A' points and 'B' points are occupied **before** the laying down of string #3 commences.

Next we lay down string #4, but since strings #1, #2 and #3 already occupy the three consecutive bight-points immediately below those which string #4 is going to occupy, we mark the bight-points immediately to the left (in cyclic fashion of course) of those who possess the letter B with C. Hence for string #4 we obtain the complementary cyclic bight-number scheme:

```
0  C  B  A  2  C  B  A  1  C  B  A
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Although the 'A' points were occupied later than the 'B' points, which in turn were later occupied than the 'C' points, this is irrelevant for our purpose since the 'A' points, 'B' points and 'C' points are occupied **before** the laying down of string #4 commences.

Finally we lay down string #5, but since strings #1, #2, #3 and #4 already oc-

/.....