Constrained Knot Representation and Its Characteristics

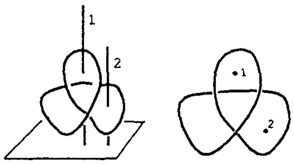
4P−11

紐デザイン処理（３）

Rahmat BUDIARTO Masashi YAMADA Hidenori ITOH Hirohisa SEKI
Nagoya Institute of Technology

1. Introduction

Characteristics of knot (string diagram) are very important in knot theory. We consider a knot in R^2 that is constrained by some points and we call it as a *constrained knot*. Fig. 1(a) shows a constrained knot diagram in space R^3 while its projection on two-dimensional plane R^2 is shown in Fig. 1(b).



(a) (b)

Fig.1 : Knot in R^3 and R^2

In this paper we develop a program for generating the polynomial of any knot and displaying its *state* diagram. The motivation of this investigation is first to express symbolically the polynomial of any knot diagram, the second is to find some basic properties of any knot diagram, and the third is to prove mathematically those properties. constrained knot diagram will be represented by several components, i.e: crosses, lines, areas, and points.

2. The Algorithm

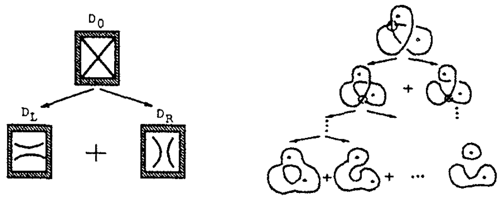
Each cross D_0 in a diagram will be transformed into D_L and D_R by using an operator in Fig. 2(a). This transformation called as *cut_and_combine* procedure. The procedure provides us a binary tree, called *resolving tree* shown in Fig. 2(b). Each of the final state diagram is obtained from the original knot diagram by using a sequence of cut_and_combine procedure recursively. The process will be terminated until all crosses are dissapeared.

2.1 Polynomial Calculation

The polynomial of constrained knot is expressed in two variables A and h as,

$$P(A^{\pm}, h) = (-A^3)^{-w(D)} \Sigma_i A^{c_i} h_{\alpha_i} < D_i >, \qquad (1)$$

$w(D)$ denotes the sum of the signs of those crossings between strands belonging to the same component and $< D >$ is the invariant of D. α_i is a sequence of the points in the region and c_i is an integer.



(a) (b)

Fig.2 : Operator for cross and resolving tree

The polynomial is calculated using Yamada's method [1]. First, the diagram is transformed into its associated graph. Then using the operator in Fig. 2(a) each edge in the graph is processed.

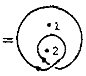
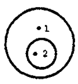
D = 

Fig.3 : Simple constrained knot

[Example 1] Calculating the polynomial of knot diagram in Fig. 3.

$w(D) = -1$

$< D >= A <$  $> +A^{-1} <$  $>$

$$P(D) = \{-A^3\}^{+1}\{A[-(A^2 + A^{-2})]h_2 h_{12} + A^{-1}[h_{12}]\}$$

2.2 The Generate Algorithm

The following is the algorithm for generating polynomial and displaying the final state diagram.

Algorithm: Generate
Begin
Input : D:knot diagram, C:cross-list and G:graph-list
Output: S:state diagrams and $P(D)$:polynomial
cut_and_combine(D, C, G)
End of algorithm Generate
Procedure cut_and_combine(D, C, G)
Begin
If all crossings are dissapeared
Display D
Calculate $P(D)$

Return

Else

Cut(D,cross-number,G,D_1)

X = cut_and_combine(D_1,C)

Combine(D,cross-number,G,D_2)

Y = cut_and_combine(D_2,C)

Case of crossing-type:

crossing-type is a: (Return $A * X + A^{-1} * Y$)

crossing-type is b: (Return $A * Y + A^{-1} * X$)

End of Case

EndIf

Print $P(D)$

End of Procedure cut_and_combine

The screen layout of the system along with an example is shown in Fig. 4 (input is in the left, output is in the right). While the polynomial is shown in Fig. 5

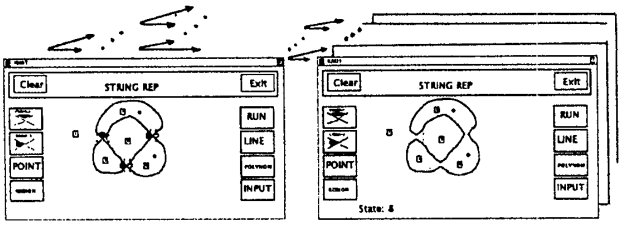


Fig.4 : Input and output example

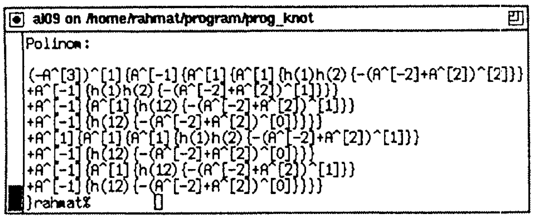


Fig.5 : The polynomial output

Lemma 1. The algorithm has time computation order $O(2^n)$ and memory space order $O(2^n)$.

3. The Properties

Definition 1.

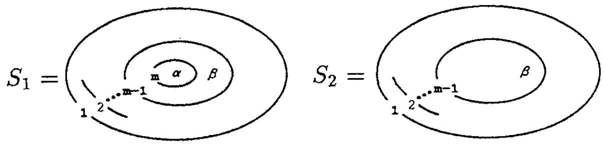
Let D be a knot diagram input to the Generate algorithm and S be its union of state diagram output . The *constrained index* of knot diagram D, is defined by

$$R(D) = \frac{I(S)}{2^{c(D)}}, \qquad (2)$$

where $c(D)$ is a number of crossing in D and $I(S)$ satisfies:

(1) I(○) = 0,   ○ is a trivial knot,

$$\text{(2) } I(S_1) = \begin{cases} \alpha + I(S_2) \text{ if } m \text{ is odd,} \\ 0 + I(S_2) \text{ if } m \text{ is even,} \end{cases}$$



(3) $I(S' \cup S'') = I(S') + I(S'')$

(4) If S is an empty set or S consists only a points then $I(S) = 0$

[Example 2] Calculating $R(D)$ (see Fig. 6).

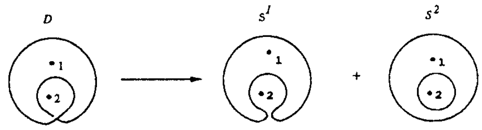


Fig.6 : Simple knot with two constrained points

$I(S) = I(S^1 \cup S^2) = (1 + 0) + (1 + 0) = 2$

$R(D) = I(S)/2^1 = 1$

The following are some lemmas about the constrained index (see Fig. 2(a)).

Lemma 2. $I(S_{D_L}) = I(S_{D_R})$

Lemma 3. $R(D_L) = R(D_R)$

Lemma 4. $R(D_0) = R(D_L) = R(D_R)$.

Lemma 5. The constrained index R is invariant under the three kinds of Reidemiester moves.

Theorem 6. If $K_1 \sim K_2$ then $R(K_1) = R(K_2)$. Constrained index is invariant of ambient isotopy.

Proof. From lemma 2 through lemma 5. □

4. Conclusion

A program for generating invariant polynomial of constrained knot diagram has been developed. This program also displays the final state diagrams. Some results lead us into conclusion that there exist a connection between a number of cross and its number of point in a diagram, called as a constrained index, and some lemmas have been proved mathematically.

References

[1] M.Yamada, H.Itoh, H.Seki, R.Budiarto, A Method of Characterization for Constrained Knot with Point, Information Processing Society of Japan.3-13 (1993)

[2] L.H. Kauffman, On Knots, Annals of Mathematics Studies. Vol. 155 (Princeton Univ. Press, Princeton, 1987).

[3] J. Hoste and J.H. Przytycki, An Invariant of Dichromatic Links, Pro. Am. Math. Soc. Vol 105, Num 4 (1989) 1003-1007.

[4] J. Hoste, A Polynomial Invariant of Knot and Link, Pac. J. Math. Vol 124, Num 2 (1986) 295-320.