〈국문초록〉

특정 가상 고리 매듭의 풀림 매듭열과 풀림수에 대한 탐구

연구자 : 이수보(2학년, suboo00@naver.com)
피진욱(2학년, jwpee@naver.com)
허준영(2학년, joeheomail@naver.com)

책임지도자 : 김훈(KAIST 부설 한국과학영재학교, HunKim@kaist.ac.kr)
공동지도자 : 이승익(KAIST 부설 한국과학영재학교, maick@hanmail.net)

요약문

이 보고서는 (p,q)-torus knot에서 마지막 한 줄을 제외한 모든 줄의 crossing을 virtual crossing으로 바꾼 $VT_{p,q}^{q-1}$의 unknotting number에 대해 알아본다. Unknotting number는 최소 몇 번의 crossing change만으로 knot을 unknot으로 만들 수 있는지를 의미한다. 이 값의 하한은 P-invariant를 이용하여 $(p-\gcd(p,q-1))/2$임을 보였고, 상한은 대략적인 unknotting sequence를 찾아내어 약간의 가정 하에 하한과 같은 값임을 밝혀내었다. 더 나아가, 이 보고서에서는 crossing change 대신 virtualization만 허용되는 경우의 unknotting number의 상한과 (p,q)-torus knot에서 마지막 두 줄을 제외한 모든 줄의 crossing이 virtual crossing인 $VT_{p,q}^{q-2}$의 unknotting number의 하한도 알아본다.

주제어 : 가상 토러스 매듭, 풀림수, 풀림 매듭열, 엇갈림 변환, 가상화

Ⅰ. 서론

어떤 knot이 unknot인지 확인하는 것은 knot theory에서 가장 기초적이고 중요한 문제다. 이를 확인하기 위해 여러 가지 방법들이 고안되었는데, 그 중 하나는 knot invariant다. Knot invariant는 동일한 knot에 대하여 변하지 않는 값으로 어떤 knot이 unknot과 같은지를 판별하는데 중요한 역할을 한다. 이러한 Knot invariant중 가장 중요한 것들 중 하나는 unknotting number다. Unknotting number는 crossing change나 virtualization을 통해 knot을 변형하여 unknot으로 만들 수 있을 때 이러한 변형의 최소 회수를 말한다. 어떤 knot의 unknotting number를 알면 unknot과 같은지 판별할 수 있기 때문에 unknotting number를 구하는 것은 knot theory에서 중요한 문제이다. 그래서 우리는 virtual knot diagram의 unknotting number, 특히 virtual torus knot의 unknotting number를 구하여 virtual torus knot의 성질에 대해 알아보고자 한다.

Ⅱ. 선행 연구

1. 가상 고리 다이어그램(Virtual link diagram)

가상 고리 다이어그램은 평면 위에 그려진 n개의 닫힌곡선으로 다음 네 가지 성질을 만족한다.

① 각 곡선은 매끄러운 곡선이며, 교차점에서는 곡선이 횡단한다.

② 교차점은 두 개의 선분으로 이루어져 있다.

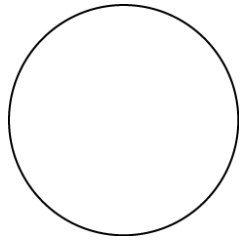
③ 각 곡선은 유한개의 직선 성분으로 근사할 수 있다.

④ 각 교차점들은 엇갈림(classical crossing) 또는 가상 엇갈림(virtual crossing)이다. 엇갈림은 위로 지나가는 선분과 아래로 지나가는 선분으로 이루어져 있으며, 가상 엇갈림은 위아래가 없는 두 선분으로 이루어져 있다.

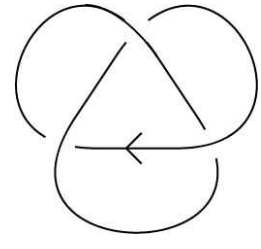
1개의 닫힌곡선으로 이루어진 가상 고리 다이어그램은 특별히 가상 매듭 다이어그램(virtual knot diagram)이라 한다. 가상 고리(Virtual link)는 가상 매듭 다이어그램의 동류항이고, 가상 매듭(virtual knot)은 가상 매듭 다이어그램의 동류항이다. 아무 엇갈림이 없는 가상 매듭 다이어그램에 대응되는 가상 매듭을 풀린매듭(unknot)이라 한다.

[그림 1] 엇갈림(좌)과 가상 엇갈림(우)

[그림 3]과 같이 가상 고리 다이어그램에서 모든 곡선들에 대해 각각 방향(orientation)을 표시한 것을 방향 가상 고리 다이어그램(oriented virtual link diagram)이라 한다.



[그림 2] 풀린매듭



[그림 3] 방향 가상 고리 다이어그램

마찬가지로, 가상 매듭 다이어그램에 방향을 표시한 것을 방향 가상 매듭 다이어그램이라 한다. 방향 가상 고리 다이어그램에서는 각 엇갈림들에 [그림 4]와 같이 부호(sign)를 부여할 수 있다. +1을 줄여서 +로, −1을 줄여서 −로도 표시한다. 앞으로 가상 고리(매듭)는 가상 고리(매듭) 다이어그램과 그 동류항을 통칭한다.

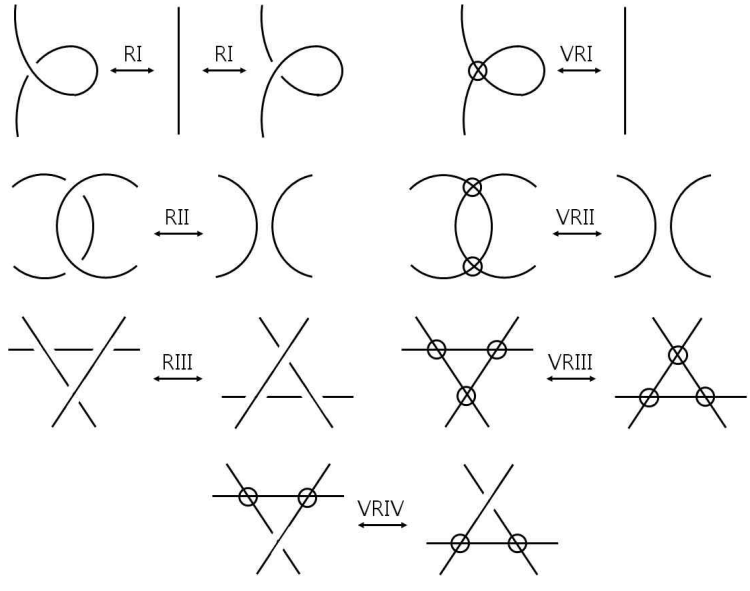
[그림 4] 부호가 +1인 엇갈림(좌)과 −1인 엇갈림(우)

2. 확장 라이데마이스터 변환(Generalized Reidemeister Moves)

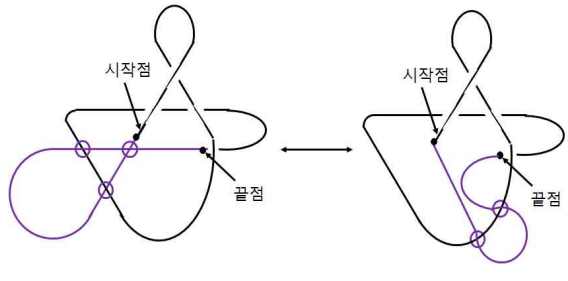
[그림 5]에서 주어진 7가지 변환을 가상 고리 다이어그램의 확장 라이데마이스터 변환이라고 한다. 그 중 RI, RII, RIII를 일반적인 라이데마이스터 변환(classical Reidemeister moves)이

라고 하고, VRI, VRII, VRIII, VRIV를 가상 라이데마이스터 변환(virtual Reidemeister moves)이라고 한다. 두 개의 가상 고리 다이어그램이 순차적인 확장 라이데마이스터 변환들에 의해서 같아질 때 두 다이어그램이 동치라고 한다. 또한, 두 가상 고리 다이어그램 P와 Q가 동치이면, $P \sim Q$로 표기한다.

우회 변환(Detour move)은 여러 번의 확장 라이데마이스터 변환을 한꺼번에 시행한 것으로, 다이어그램 위의 두 점 사이에 엇갈림이 없는 부분을 선택하여 그 두 점을 자유로운 형태로 연결하고, 기존의 다이어그램과 새로 만나는 부분은 모두 가상 엇갈림으로 표시하는 변환이다.



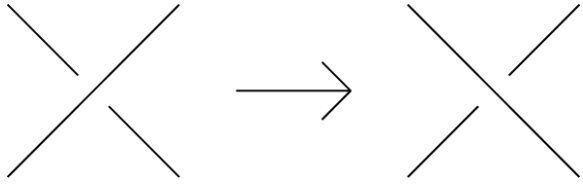
[그림 5] 확장 라이데마이스터 변환



[그림 6]우회 변환

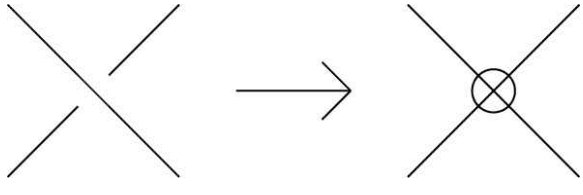
3. 가상 풀림수(Virtual Unknotting Number)

엇갈림 변환(crossing change)은 엇갈림에서 위로 지나가는 선분과 아래로 지나가는 선분의 위치를 바꾸는 변환이다. 가상 매듭 K를 풀린 매듭으로 만드는데 필요한 최소한의 엇갈림 변환의 횟수를 가상 풀림수 $vu(K)$라고 한다. 유한 번의 엇갈림 변환으로 풀린 매듭으로 만들 수 없을 때, $vu(K)=\infty$로 표시한다.



[그림 7] 엇갈림 변환

가상화(Virtualization)는 엇갈림을 가상 엇갈림으로 바꾸는 변환이다. 가상화 풀림수(virtualization unknotting number) $vvu(K)$는 가상 매듭 K를 풀린 매듭으로 만드는데 필요한 최소한의 가상화 횟수이다. 엇갈림 변환 또는 가상화를 통해 주어진 가상 매듭을 풀린 매듭으로 만들어가는 과정을 풀림 매듭열(unknotting sequence)이라고 한다.



[그림 8] 가상화

4. 땋임(Braid)

n가닥 땋임 군(n-strand braid group)은 생성자(generator) $\{\sigma_i | 1 \le i \le n-1\}$와 $\{v_i | 1 \le i \le n-1\}$로 구성되어 있고, 다음의 조건을 만족한다.

① 고전 관계(classical relations)

㉮ $\sigma_i\sigma_j = \sigma_j\sigma_i (|i-j| \ge 2)$

㉯ $\sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1}$ (RIII)

② 가상 관계(virtual relations)

㉮ $v_i^2 = 1$(VRII)

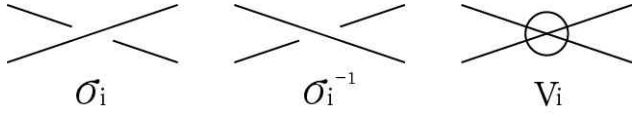
㉯ $v_i v_j = v_j v_i (|i-j| \geq 2)$

㉰ $v_i v_{i+1} v_i = v_{i+1} v_i v_{i+1}$ (VRIII)

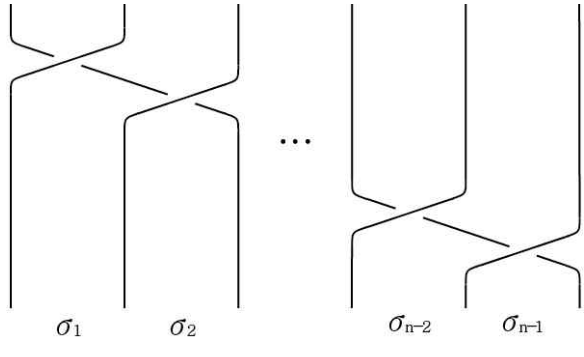
③ 혼합 관계(mixed relations)

㉮ $\sigma_i v_j = v_j v_i (|i-j| \geq 2)$

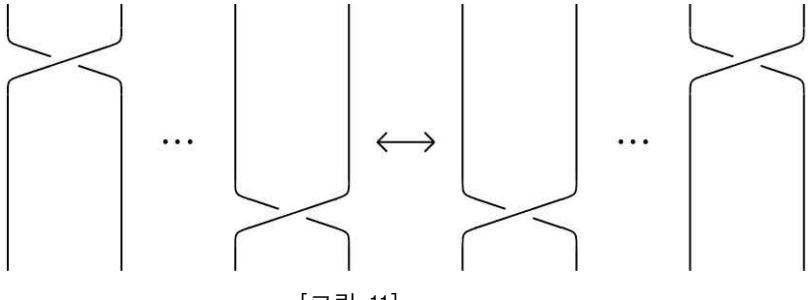
㉯ $v_i \sigma_{i+1} v_i = v_{i+1} \sigma_i v_{i+1}$ (VRIV)



[그림 9] σ_i는 i번째와 $i+1$번째 가닥의 엇갈림을 나타내는 생성자이다.

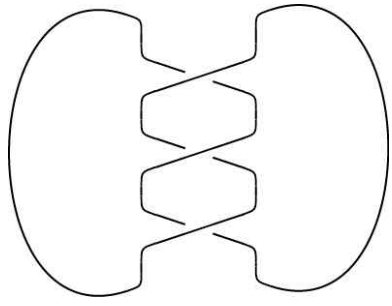


[그림 10] 땋임



[그림 11] $\sigma_i \sigma_j = \sigma_j \sigma_i$

닫힌 땋임(Closed braid)은 땋임의 각 가닥의 위와 아래를 연결한 것으로 매듭이나 고리가 된다. 이 연구에서는 편의상 땋임을 왼쪽에서 오른쪽 방향으로 표현하였으며, 땋임은 닫힌 땋임과 혼용하여 쓰인다.



[그림 12] 세잎매듭(Trefoil)(σ_1^3)

5. 가우스 문자열(Gauss Word)

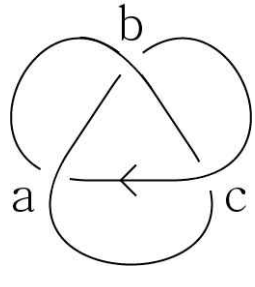
가우스 문자열은 매듭을 표현하는 문자열이다. 방향 매듭에서 가우스 문자열은 다음과 같은 방법으로 만든다.

① 매듭 위에 임의의 시작점을 선택한다.

② 엇갈림에 서로 다른 기호를 부여한다.

③ 시작점에서부터 매듭의 방향을 따라 이동하며 엇갈림을 만나면 그 엇갈림의 기호를 쓴다. 이 때, 엇 갈림의 부호를 기호 뒤에 붙인다. 또한, 그 엇갈림을 위로 지나갈 때 기호 위에 ¯를 붙인다.

④ 시작점으로 다시 돌아올 때까지 위의 과정을 반복한다.



[그림 13] 세잎매듭

[그림 13]의 매듭은 $a+\bar{b}+c+\bar{a}+b+\bar{c}+$을 가우스 문자열로 가진다.

6. 가우스 다이어그램(Gauss Diagram)

가우스 다이어그램은 가우스 문자열을 원 위에 표시한 것으로, 임의의 방향 가상 매듭 K에 대한 가우스 다이어그램 $G(K)$는 다음과 같이 만든다.

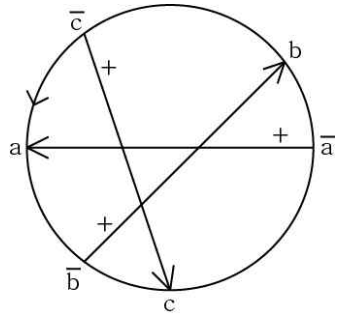
① 원주 위에 임의의 시작점을 잡는다.

② 시작점에서 시계 반대 방향으로 K의 가우스 문자열의 기호들을 원주 위에 순서대로 적는다.

③ 원주 위의 같은 기호들을 화살표로 잇는다. 이때 화살표의 시작점은 ¯가 붙어있는 기호이고, 끝점은 ¯가 없는 기호이다. 화살표를 그릴 때 서로 다른 세 화살표가 한 점에서 만나지 않도록 조정한다.

④ 각 화살표에 그 기호에 대응하는 부호를 붙인다.

앞으로 $G(K)$를 방향 가상 매듭 K의 가우스 문자열과 가우스 다이어그램으로 혼용하겠다.



[그림 14] 세잎매듭의 가우스 다이어그램

7. P-불변량(P-Invariant)

P-불변량는 방향 가상 매듭의 가우스 다이어그램서 계산할 수 있는 불변량이다. 우선, 임의의 방향 가상 매듭 K의 가우스 다이어그램 $G(K)$에 대해 $G(K)$의 모든 화살표들의 집합을 $A(G(K))$라고 하자. 또한, $a\in A(G(K))$에 대해 a의 시작점에서 출발하여 시계 반대 방향으로 이동하여 끝점에 도착할 때까지 이동한 원호를 γ_1이라고 하고 나머지 부분의 원호를 γ_2라고 하자. 그리고 $a\in A(G(K))$에 대해 $i_i(a)$, $i_o(a)$, $i(a)$를 다음과 같이 정의한다.

$$i_i(a) := \{c \in A(G(K)) | c \neq a, c\text{의 시작점이 } \gamma_2, \text{끝점이 } \gamma_1 \text{ 위에 있다.}\}$$
$$i_o(a) := \{c \in A(G(K)) | c \neq a, c\text{의 시작점이 } \gamma_1, \text{끝점이 } \gamma_2 \text{ 위에 있다.}\}$$
$$i(a) := \sum_{c \in i_i(a)} sign(c) - \sum_{c \in i_o(a)} sign(c)$$

정의 2.1. 임의의 방향 가상 매듭 K에 대해 $P(K)$를 다음과 같이 정의한다.

$$P(K) := \sum_{\substack{c \in A(G(K)) \\ i(c) \neq 0}} sign(c) t^{|i(c)|}$$

이를 **P-불변량**이라고 한다.

정리 2.2.[1] P-불변량은 매듭 불변량(knot invariant), 즉 확장된 라이데마이스터 변환에 대해 불변이다.

정리 2.3.[1] P-불변량은 엇갈림 변환에 대해 불변이다.

정리 2.4.[1] 방향 가상 매듭 K의 P-불변량이 $P(K) = \sum_{m \in \mathbb{N}} b_m t^m$ 일 때

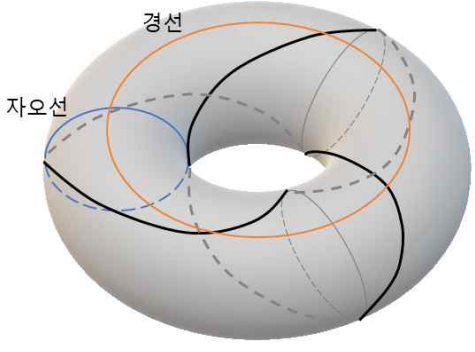
$$vu(K) \geq \frac{1}{2} \sum_{m \in \mathbb{N}} |b_m|$$

이다.

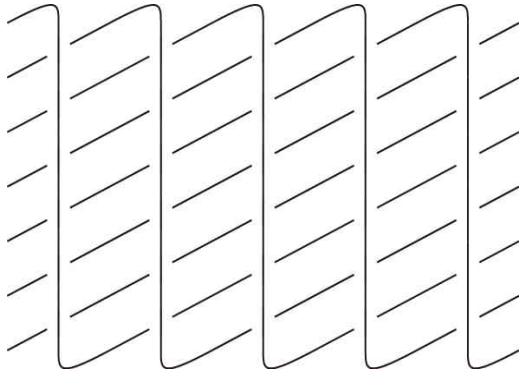
8. 가상 토러스 매듭(virtual torus knot) $VT^n_{p,q}$의 정의

정의 2.5. (p,q)**-토러스 고리**는 토러스 위에 놓인 단순 닫힌곡선으로, 자오선(meridian) 방향으로 p회, 경선(longitude) 방향으로 q회 회전하는 고리이다. 다음 그림에서 m은 자오선이고, l은 경선이다.

(p,q)-토러스 매듭(torus knot)은 성분 1개인 토러스 고리로, (p,q)-토러스 고리는 $\gcd(p,q) = 1$이면 (p,q)-토러스 매듭이 된다. 또한, (p,q)-토러스 고리는 [그림 16]과 같이 q개 줄의 위로 지나가는 가닥을 가진 p-가닥 땋임으로 나타낼 수 있으며, 이를 토러스 고리의 땋임형(braid form)이라고 한다.

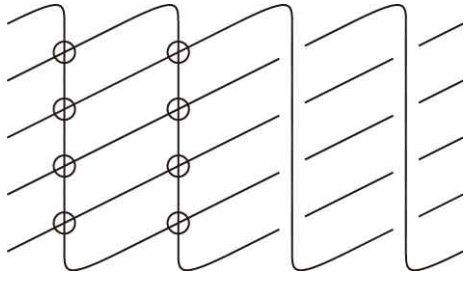


[그림 15] $(2,3)$-토러스 매듭



[그림 16] $(7,5)$-토러스 매듭의 땋임형

(p,q)-토러스 매듭의 땋임형에서 $1\sim n$번째 위로 지나가는 가닥의 엇갈림들을 가상화시킨 것을 $VT^n_{p,q}$로 표시한다.



[그림 17] $VT^2_{5,4}$

9. 가상 토러스 매듭 $VT^n_{p,q}$의 가상 풀림수 $vu(VT^n_{p,q})$의 하계

집합 A_0와 A_1을 다음과 같이 정의하자.

$$A_0 = \{c \in A(G(VT^n_{p,q}) | i(c) = 0\}$$
$$A_1 = \{c \in A(G(VT^n_{p,q}) | i(c) \neq 0\}$$

정리 2.6.[1] p와 n이 서로소인 가상 토러스 매듭 $VT^n_{p,q}$의 가우스 다이어그램 $G(VT^n_{p,q})$에서 A_0는 공집합이다. 즉, $A(G(VT^n_{p,q}))$은 A_1과 같다.

정리 2.7.[1] p와 n이 서로소인 가상 토러스 매듭 $VT^n_{p,q}$의 가상 풀림수의 하계는 다음과 같다.

$$vu(VT^n_{p,q}) \geq \frac{(p-1)(q-n)}{2}$$

정리 2.8.[1] 가상 토러스 매듭 $VT^n_{p,q}$의 가상 풀림수의 하계는 다음과 같다.

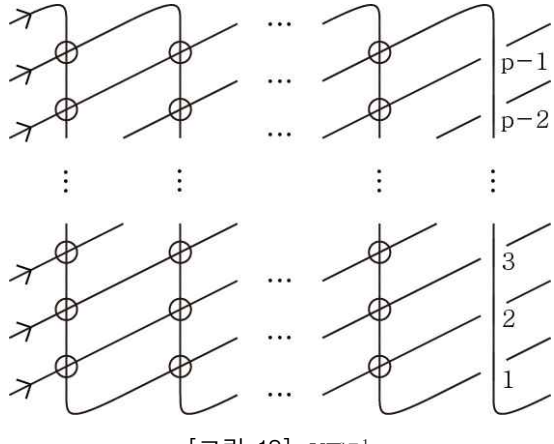
$$vu(VT^n_{p,q}) \geq \frac{(p-1)(q-n)-|A_0|}{2}$$

증명) $\sum_{m\in\mathbb{N}} |b_m| = |A_1| = |A(G(VT^n_{p,q}))| - |A_0| = (p-1)(q-n) - |A_0|$. 따라서 정리 2.3에 의해 이 정리가 성립한다. □

따라서 $|A_0|$를 알면 $vu(VT^n_{p,q})$의 하계를 구할 수 있다. 또한, 임의의 가상 토러스 매듭의 엇갈림들에 대한 P-불변량을 알면 $|A_0|$를 구할 수 있다.

Ⅲ. 연구내용

1. 가상 토러스 매듭 $VT^{q-1}_{p,q}$의 가상 풀림수 $vu(VT^{q-1}_{p,q})$의 하계



[그림 18] $VT^{q-1}_{p,q}$

[그림 18]과 같이 주어지는 $VT_{p,q}^{q-1}$의 가우스 문자열은 다음과 같다.

$$\overline{p-1}+\overline{p-2}+...+\overline{2}+\overline{1}+\sum_{i=1}^{p-1}(qi \bmod p)+$$

여기서 $\sum$는 기호들을 순서대로 나열하는 의미이다.

방향 가상 매듭 K의 가우스 다이어그램에서의 화살표 $a\in A(G(K))$에 대하여 $I(a)$와 $O(a)$를 다음과 같이 정의한다.

$$I(a)=\{c\in A(G(K))|c\neq a, c\text{의 끝점이 }\gamma_1\text{위에 있다.}\}$$
$$O(a)=\{c\in A(G(K))|c\neq a, c\text{의 시작점이 }\gamma_1\text{위에 있다.}\}$$

보조정리 3.1. 방향 가상 매듭 K의 엇갈림의 부호가 모두 $+$일 때, 가우스 다이어그램에서의 화살표 $a\in A(G(K))$에 다음이 성립한다.

$$i(a)=|I(a)|-|O(a)|$$

증명) 시작점이 γ_2 위에 있고 끝점이 γ_1 위에 있는 화살표의 개수는

$$|I(a)-I(a)\cap O(a)|$$

이고, 시작점이 γ_1 위에 있고 끝점이 γ_2 위에 있는 화살표의 개수는

$$|O(a)-I(a)\cap O(a)|$$

이다. 따라서 $i(a)$의 정의에 의해

$$i(a)=|I(a)-I(a)\cap O(a)|-|O(a)-I(a)\cap O(a)|=|I(a)|-|O(a)|$$

이다. □

정리 3.2. 가상 토러스 매듭 $VT_{p,q}^{q^{-1}}$의 $|A_0| = \gcd(p, q-1) - 1$ 이다.

증명) $VT_{p,q}^{q^{-1}}$의 가우스 문자열 중 $\overline{k}+, k+$이 $A(G(VT_{p,q}^{q^{-1}}))$의 원소 a에 대응된다고 하자. (단, $1 \le k \le p-1$이다.) 이 때 $rq \bmod p = k$를 만족하는 가장 작은 자연수를 r로 놓으면

$$O(a) = \{\overline{1}+, \overline{2}+, \ldots, \overline{k-1}+\},$$
$$I(a) = \{q \bmod p+, 2q \bmod p+, \ldots, (r-1)q \bmod p+\}$$

가 된다. 따라서 $|O(a)| = k-1$, $|I(a)| = r-1$가 되므로 보조정리 3.1에 의해 $i(a) = 0$인 것과 $r = k$인 것이 동치가 된다. 또한 $1 \le k \le p-1$이기 때문에

$$r = k \Leftrightarrow kq \bmod p = k$$
$$\Leftrightarrow k(q-1) \equiv 0 \pmod p$$
$$\Leftrightarrow \text{어떤 자연수 } n\text{에 대해 } k = \frac{np}{q-1}$$

가 된다. 따라서 이 식을 만족하는 $1 \le k \le p-1$ 범위 내에 있는 자연수 k의 개수가 $i(a) = 0$인 지점의 개수와 같다.

$a = \gcd(p, q-1)$로 놓으면,

$$k = \frac{np/a}{(q-1)/a}$$

가 되고, $\frac{p}{a}$와 $\frac{q-1}{a}$가 서로소이므로 k가 자연수가 되기 위해서는 어떤 자연수 α에 대해

$$n = \frac{\alpha(q-1)}{a}$$

이다. 따라서

$$k = \frac{\alpha p}{a}$$

이다. 그리고 $1 \le k \le p-1$이므로 $1 \le \alpha \le a-1$이다. 또한, k는 α에 의해 유일하게 결정되므로

$$|A_0| = a-1 = \gcd(p,q-1)-1$$

이다. □

위 사실을 종합하면 다음이 성립한다.

정리 3.3. 가상 토러스 매듭 $VT_{p,q}^{q-1}$의 가상 풀림수 $vu(VT_{p,q}^{q-1})$의 하계는 다음과 같다.

$$vu(VT_{p,q}^{q-1}) \ge \frac{p-1}{2} - \frac{\gcd(p,q-1)-1}{2}$$

증명) 정리 2.8과 정리 3.2에 의해 성립한다. □

2. 가상 토러스 매듭 $VT_{p,q}^{q-1}$의 가상 풀림수 $vu(VT_{p,q}^{q-1})$의 상계

정의 3.4. 다음을 만족하는 유한한 자연수 수열 $\{a_i\}$들을 **M-수열**이라고 한다.

① $r = |\{a_i\}|$

② $a_i \ne a_j,\ \forall i \ne j$

③ $1 \le a_i \le r,\ \forall 1 \le i \le r$

④ $a_i + a_{r-i+1} = r+1,\ \forall 1 \le i \le r$

⑤ $a_{k+i} = a_i + 1,\ \forall 1 \le i \le r-k$, where $a_k = 1$

⑥ $a_1 \ne 1$

그리고, 조건 ①~⑤까지 만족하고 $a_1 = 1$인 자연수 수열 $\{a_i\}$들을 **M1-수열**이라고 한다.

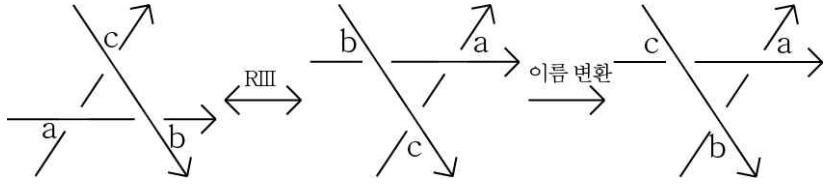
정의 3.5. 다음을 만족하는 가우스 문자열 $G(K)$들을 **M-문자열**이라고 한다.

$$G(K) = \overline{r} + \overline{r-1} + \ldots + \overline{2} + \overline{1} + \sum_{i=1}^{r} a_i + \text{ (단, } \{a_i\}\text{는 M-수열)}$$

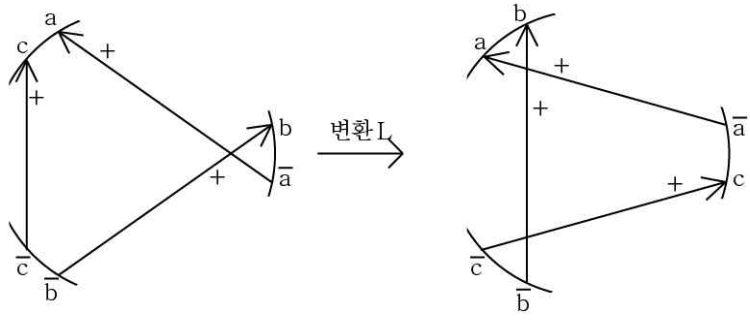
이후 $\{a_i\}$는 그에 대응되는 M-문자열과 혼용한다. M1-문자열도 이와 비슷하게 정의한다.

정의 3.6. 다음 [그림 19]와 같이 RIII 변환을 한 후 b와 c의 이름을 바꿔주는 과정을 변환 L이라고 한다. 변환 L을 가우스 다이어그램으로 나타내면 [그림 20]과 같다. 결과적으로, 변환 L에서는 a와 $\overline{a}$가 반시계방향으로 움직였고, 반대로 b와 c는 시계

방향으로 움직인다. 이 변환을 (a, b, c)라고도 하겠다.



[그림 19] 변환 L



[그림 20] 가우스 다이어그램에서의 변환 L

정의 3.7. 아래 그림은 한 번의 엇갈림 변환과 RII 변환을 이용하여 두 엇갈림을 풀어줄 때 나타나는 가우스 다이어그램의 변화를 나타낸 것이다. 이 때 가우스 문자열에서는 a, b, $\bar{a}$와 $\bar{b}$가 지워지게 되며, 이 변환을 변환 P라고 한다. 이 변환을 (a, b)라고도 하겠다.

[그림 21] 변환 P

정리 3.8. M-문자열 $\{a_i\}$에서 $a_k = 1$인 자연수 k에 대해 $x = [\frac{r}{k}]$라고 하면, $\{a_i\}$는

$$\bar{r}+\ldots+\bar{2}+\bar{1}+\sum_{i=1}^{k-1}a_i+1+\sum_{i=1}^{k-1}(a_i+1)+2+\ldots+x+\sum_{i=1}^{r-kx}(a_i+x)+,\ k\neq 1$$

꼴로 나타난다.

증명) M-문자열의 정의에 의해 $\{a_i\}$는

$$\overline{r}+\overline{r-1}+...+\overline{2}+\overline{1}+\sum_{i=1}^{r}a_i+$$

과 같이 표현된다. 또한, M-수열의 정의 중 조건 ⑤에 의해

$$a_{kj+i}=a_{k(j-1)+i}+1=a_{k(j-2)+i}+2=...=a_i+j\ (1\le i\le k-1,\ j\in \mathbf{N})$$

이 된다. 또한, M-수열의 정의 중 조건 ⑥에 의해

$$k\neq 1$$

이 성립한다. 이 사실들을 종합하면 $\{a_i\}$가

$$\overline{r}+...+\overline{2}+\overline{1}+\sum_{i=1}^{k-1}a_i+1+\sum_{i=1}^{k-1}(a_i+1)+2+...+x+\sum_{i=1}^{r-kx}(a_i+x)+,\ k\neq 1$$

로 표현된다. □

정의 3.9. M-문자열 $\{a_i\}$가 정리 3.8에서와 같이 나타날 때, 이 문자열을 다음과 같이 바꾸는 변환을 변환 H라고 한다.

$$\overline{r-2x}+\overline{r-2x-1}+...+\overline{2}+\overline{1}+\sum_{i=1}^{r-kx}(a_i)+\sum_{i=r-kx+2}^{k-1}(a_i-1)+\sum_{i=1}^{r-kx}(a_i-1)+$$
$$\sum_{i=r-kx+2}^{k-1}(a_i-2)+...+\sum_{i=1}^{r-kx}(a_i-(x-1))+\sum_{i=r-kx+2}^{k-1}(a_i-x)+\sum_{i=1}^{r-kx}(a_i-x)+$$

정리 3.10. 변환 H는 유한 번의 변환 L와 변환 P를 사용하여 나타낼 수 있으며, 이 때 변환 c에 의해서 없어지는 엇갈림의 개수는 변환 H 안에 포함되어 있는 엇갈림 변환의 개수의 두 배와 같다.

증명) 우선, $a_k=1$인 자연수 k에 대해 $x=[\frac{r}{k}]$로 놓자. 정리 3.9에 의해 $\{a_i\}$는

$$\overline{r}+\ldots+\overline{2}+\overline{1}+\sum_{i=1}^{k-1}a_i+1+\sum_{i=1}^{k-1}(a_i+1)+2+\ldots+x+\sum_{i=1}^{r-kx}(a_i+x)+,\ k\neq 1$$

꼴로 나타난다. 이 가우스 문자열을 변환 L과 변환 P를 유한 번 시행하여 정의 3.10에 있는 가우스 문자열로 변환시킬 것이다.

과정 1-1-1. $\sum_{i=1}^{k-1}a_i+$와 $\sum_{i=1}^{k-1}(a_i+1)+$의 각 항이 정확히 1씩 차이나고, $\overline{r}+\ldots+\overline{2}+\overline{1}+$ 부분의 항들이 1씩 감소하기 때문에 $(1,\ a_1,\ a_1+1)$, $(1,\ a_2,\ a_2+1)$, …, $(1,\ a_{k-1},\ a_{k-1}+1)$을 순차적으로 시행해줄 수 있고, 이를 시행하면

$$\overline{r}+\ldots+\overline{2}+\sum_{i=1}^{k-1}(a_i+1)+\overline{1}+\sum_{i=1}^{k-1}a_i+1+2+\sum_{i=1}^{k-1}(a_i+2)+\ldots+x+\sum_{i=1}^{r-kx}(a_i+x)+$$

가 된다.

과정 1-1-2. $\sum_{i=1}^{k-1}(a_i+1)+$와 $\sum_{i=1}^{k-1}(a_i+2)+$의 각 항이 정확히 1씩 차이나고, $\overline{r}+\ldots+\overline{2}+$ 부분의 항들이 1씩 감소하기 때문에 $(2,\ a_1+1,\ a_1+2)$, $(2,\ a_2+1,\ a_2+2)$, …, $(2,\ a_{k-1}+1,\ a_{k-1}+2)$을 순차적으로 시행해줄 수 있고, 이를 시행하면

$$\overline{r}+\ldots+\overline{3}+\sum_{i=1}^{k-1}(a_i+2)+\overline{2}+\overline{1}+\sum_{i=1}^{k-1}a_i+1+\sum_{i=1}^{k-1}(a_i+1)+2+3+\ldots+x+\sum_{i=1}^{r-kx}(a_i+x)+$$

가 된다.

과정 1-1-1과 1-1-2와 비슷하게 과정 1-1-3, 1-1-4, …, 1-1-$(x-1)$도 아래와 같이 시행한다.

과정 1-1-y. $(3\leq y\leq x-1)$ $\sum_{i=1}^{k-1}(a_i+(y-1))+$과 $\sum_{i=1}^{k-1}(a_i+y)+$의 각 항이 정확히 1씩 차이나고, $\overline{r}+\ldots+\overline{y}+$ 부분의 항들이 1씩 감소하기 때문에 $(y,\ a_1+(y-1),\ a_1+y)$, $(y,\ a_2+(y-1),\ a_2+y)$, …, $(y,\ a_{k-1}+(y-1),\ a_{k-1}+y)$을 순차적으로 시행해줄 수 있고, 이를 시행하면

$$\overline{r}+\ldots+\overline{y+1}+\sum_{i=1}^{k-1}(a_i+y)+\overline{y}+\ldots+\overline{1}+\sum_{i=1}^{k-1}a_i+1+\sum_{i=1}^{k-1}(a_i+1)+2+\sum_{i=1}^{k-1}(a_i+2)+\ldots+x+\sum_{i=1}^{r-kx}(a_i+x)+$$

가 된다.

과정 1-1-$(x-1)$이 끝난 후 가우스 문자열은 다음과 같이 표현된다.

$$\overline{r}+..+\overline{x}+\sum_{i=1}^{k-1}(a_i+(x-1))+\overline{x-1}+..+\overline{1}+\sum_{i=1}^{k-1}a_i+1+\sum_{i=1}^{k-1}(a_i+1)+2+\sum_{i=1}^{k-1}(a_i+2)+..+x+\sum_{i=1}^{r-kx}(a_i+x)+$$

과정 1-1-x. $\sum_{i=1}^{r-kx}(a_i+(x-1))+$과 $\sum_{i=1}^{r-kx}(a_i+x)+$의 각 항이 정확히 1씩 차이나고, $\overline{r}+...+\overline{y}+$ 부분의 항들이 1씩 감소하기 때문에 $(x,\ a_1+(x-1),\ a_1+x)$, $(x,\ a_2+(x-1),\ a_2+x)$, …, $(y,\ a_{r-kx}+(x-1),\ a_{r-kx}+x)$을 순차적으로 시행해줄 수 있고, 이를 시행하면

$$\overline{r}+..+\overline{x+1}+\sum_{i=1}^{r-kx}(a_i+x)+\overline{x}+\sum_{i=r-kx+1}^{k-1}(a_i+(x-1))+\overline{x-1}+...+\overline{1}+$$
$$\sum_{i=1}^{k-1}a_i+1+\sum_{i=1}^{k-1}(a_i+1)+2+\sum_{i=1}^{k-1}(a_i+2)+..+(x-1)+\sum_{i=1}^{r-kx}(a_i+(x-1))+x+$$

가 된다.

과정 1-2. 과정 1-1-x 후에 나온 가우스 문자열 중

$$\overline{x-1}+..+\overline{1}+\sum_{i=1}^{k-1}a_i+1+\sum_{i=1}^{k-1}(a_i+1)+2+\sum_{i=1}^{k-1}(a_i+2)+..+(x-1)+\sum_{i=1}^{r-kx}(a_i+(x-1))+$$

부분을 보면 과정 1-1 전반에서 한 시행을 비슷하게 시행해줄 수 있음을 알 수 있다. 단, 이 때에는 과정 1-1에서와는 다르게 1, 2, …, x에 대해서 시행해주는 것이 아니라 1, 2, …, $x-1$에 대해서만 시행하게 된다. 이를 시행하면 아래와 같이 된다.

$$\overline{r}+..+\overline{x+1}+\sum_{i=1}^{r-kx}(a_i+x)+\overline{x}+\sum_{i=r-kx+1}^{k-1}(a_i+(x-1))+$$
$$\sum_{i=1}^{r-kx}(a_i+(x-1))+\overline{x-1}+\sum_{i=r-kx+1}^{k-1}(a_i+(x-2))+\overline{x-2}+...+\overline{1}+$$
$$\sum_{i=1}^{k-1}a_i+1+\sum_{i=1}^{k-1}(a_i+1)+2+\sum_{i=1}^{k-1}(a_i+2)+..+(x-2)+\sum_{i=1}^{r-kx}(a_i+(x-2))+(x-1)+x+$$

과정 1-2와 비슷하게 과정 1-3, 1-4, …, 1-x도 아래와 같이 시행한다.

과정 1-z. ($3 \le z \le x$) 과정 1-($z-1$) 후에 나온 가우스 문자열 중

$$\overline{x-(z-1)}+..+\overline{1}+\sum_{i=1}^{k-1}a_i+1+\sum_{i=1}^{k-1}(a_i+1)+2+\sum_{i=1}^{k-1}(a_i+2)+..+(x-(z-1))+\sum_{i=1}^{r-kx}(a_i+(x-(z-1)))+$$

부분을 보면 과정 1-1 전반에서 한 시행을 비슷하게 시행해줄 수 있음을 알 수 있다. 단, 이 때에는 과정 1-1에서와는 다르게 1, 2, …, x에 대해서 시행해주는 것이 아니라 1, 2, …, $x-(z-1)$에 대해서만 시행하게 된다. 이를 시행하면 아래와 같이 된다.

$$\overline{r}+..+\overline{x+1}+\sum_{i=1}^{r-kx}(a_i+x)+\overline{x}+\sum_{i=r-kx+1}^{k-1}(a_i+(x-1))+$$
$$\sum_{i=1}^{r-kx}(a_i+(x-1))+\overline{x-1}+\sum_{i=r-kx+1}^{k-1}(a_i+(x-2))+$$
$$\sum_{i=1}^{r-kx}(a_i+(x-2))+\overline{x-2}+\sum_{i=r-kx+1}^{k-1}(a_i+(x-3))+$$
$$...+\sum_{i=1}^{r-kx}(a_i+(x-(z-1)))+\overline{x-(z-1)}+\sum_{i=r-kx+1}^{k-1}(a_i+(x-z))+\overline{x-z}+...+\overline{1}+$$
$$\sum_{i=1}^{k-1}a_i+1+\sum_{i=1}^{k-1}(a_i+1)+2+\sum_{i=1}^{k-1}(a_i+2)+..+(x-z)+\sum_{i=1}^{r-kx}(a_i+(x-z))+$$
$$(x-(z-1))+...+(x-1)+x+$$

과정 1-x가 끝난 후 가우스 문자열은 다음과 같이 표현된다.

$$\overline{r}+..+\overline{x+1}+\sum_{i=1}^{r-kx}(a_i+x)+\overline{x}+\sum_{i=r-kx+1}^{k-1}(a_i+(x-1))+$$
$$\sum_{i=1}^{r-kx}(a_i+(x-1))+\overline{x-1}+\sum_{i=r-kx+1}^{k-1}(a_i+(x-2))+$$
$$\sum_{i=1}^{r-kx}(a_i+(x-2))+\overline{x-2}+\sum_{i=r-kx+1}^{k-1}(a_i+(x-3))+$$
$$...+\sum_{i=1}^{r-kx}(a_i+1)+\overline{1}+\sum_{i=r-kx+1}^{k-1}a_i+1+...+(x-1)+x+$$

과정 2. 과정 1-x까지 시행한 후의 가우스 문자열에서 가장 뒷부분과 가장 앞부분을 붙여서 보면

$$1+\ldots+(x-1)+x+\overline{r}+\overline{r-1}+\ldots+\overline{r-x+1}+\ldots+\overline{x+1}$$

가 된다. $r-x+1 \geq x+1$인 이유는 $x=[\frac{r}{k}]$로 정의되는데 $k \geq 2$이기 때문이다. 또한, $\overline{w}$ $(1 \leq w \leq x)$ 오른쪽에는 항상 $(a_{r-kx+1}+(w-1))+$가 있게 되는데, M-수열의 정의에 의해 $a_i+a_{r-i+1}=r+1$이므로,

$$a_{r-kx+1}+(w-1)=(r+1)-a_{kx}+(w-1)=(r+1)-x+(w-1)=r-x+w$$

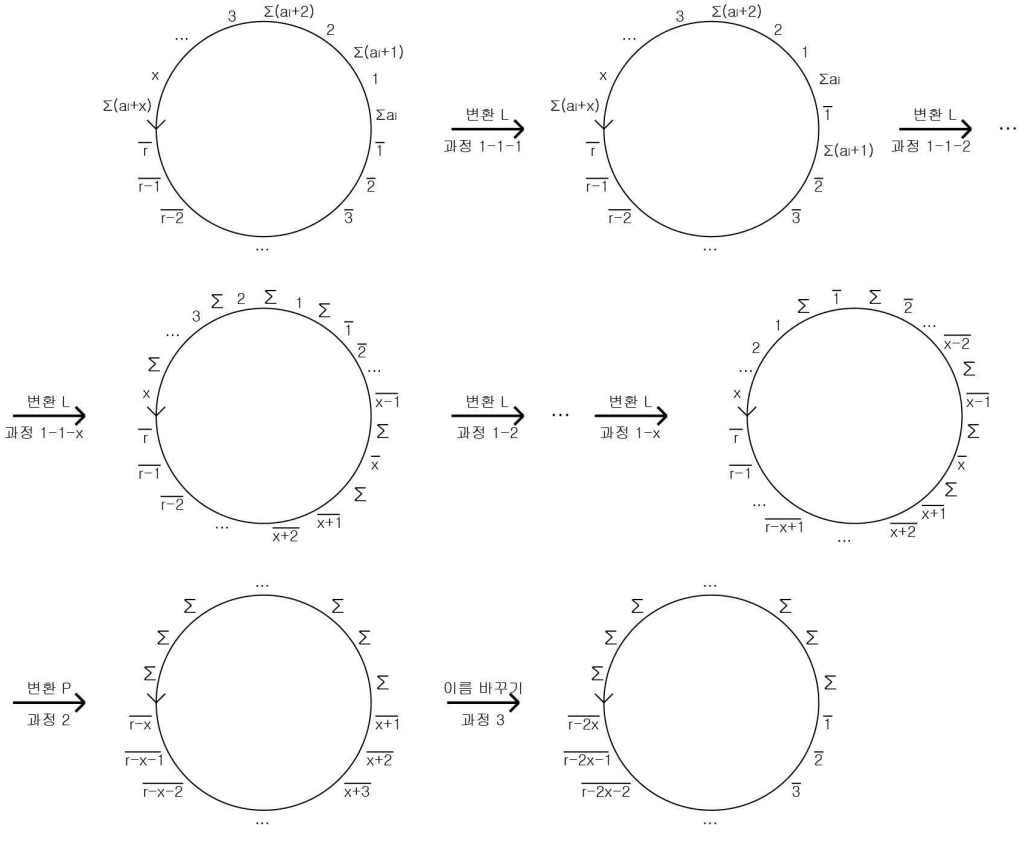
가 성립하게 된다. 따라서 $(x, r), (x-1, r-1), \ldots, (1, r-x+1)$를 순차적으로 시행해줄 수 있다. 이들을 시행하면, 가우스 문자열은 다음과 같이 표현된다.

$$\overline{r-x}+\ldots+\overline{x+1}+\sum_{i=1}^{r-kx}(a_i+x)+\sum_{i=r-kx+2}^{k-1}(a_i+(x-1))+\sum_{i=1}^{r-kx}(a_i+(x-1))+\sum_{i=r-kx+2}^{k-1}(a_i+(x-2))+$$
$$\sum_{i=1}^{r-kx}(a_i+(x-2))+\sum_{i=r-kx+2}^{k-1}(a_i+(x-3))+\ldots+\sum_{i=1}^{r-kx}(a_i+1)+\sum_{i=r-kx+2}^{k-1}a_i+$$

과정 3. 과정 2에 의해서 $1, 2, \ldots, x, \overline{1}, \overline{2}, \ldots, \overline{x}$가 없어졌기 때문에 다시 수들을 1부터 배열하기 위하여 모든 수를 x만큼씩 감소시킨다. 그러고 나면 가우스 문자열은 다음과 같다.

$$\overline{r-2x}+\ldots+\overline{1}+\sum_{i=1}^{r-kx}a_i+\sum_{i=r-kx+2}^{k-1}(a_i-1)+\sum_{i=1}^{r-kx}(a_i-1)+\sum_{i=r-kx+2}^{k-1}(a_i-2)+$$
$$\sum_{i=1}^{r-kx}(a_i-2)+\sum_{i=r-kx+2}^{k-1}(a_i-3)+\ldots+\sum_{i=1}^{r-kx}(a_i-(x-1))+\sum_{i=r-kx+2}^{k-1}(a_i-x)+$$

따라서 과정 1, 2, 3을 시행해주면 변환 L과 변환 P를 유한 번 시행하여 변환 H를 나타내줄 수 있다.

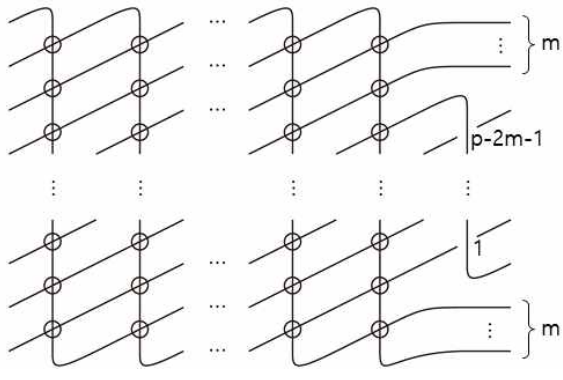


[그림 22] 과정 1, 2, 3

또한, 이 때 엇갈림은 오직 변환 P만을 통해서 사라질 수 있는데, 변환 P에서는 엇갈림 변환 한 번을 이용해 엇갈림 두 개를 사라지게 한다. 따라서 변환 H에 의해서 없어지는 엇갈림의 개수는 변환 H 안에 포함되어 있는 엇갈림 변환의 개수의 두 배와 같다. □

이후 변환 H를 정리 3.10에서의 증명 과정에서와 같이 유한 번의 변환 L과 변환 P의 시행으로 생각하겠다.

$VT_{p,q}^{q-1}$ 중 엇갈림으로만 이루어진 q번째 가닥의 위, 아래의 m개의 엇갈림을 [그림 22]와 같이 제거한 형태의 가상 토러스 고리를 $RVT_{p,q}^{m}$로 표시한다. (단, $0 \le m \le \left\lfloor \frac{p}{2} \right\rfloor$)



[그림 23] $RVT_{p,q}^{m}$

정리 3.11. 가상 토러스 고리 $RVT_{p,q}^{m}$가 매듭이고, $a_k=1$인 자연수 k가 1이 아니면, 가우스 문자열 $G(RVT_{p,q}^{m})$는 M-문자열이다. 또한, $x=\left\lfloor\frac{p-2m-1}{k}\right\rfloor$로 놓으면 변환 H에 의해서 가우스 문자열 $G(RVT_{p,q}^{m})$는 $G(RVT_{p,q}^{m+x})$으로 바뀐다. 반면 $k=1$이면 $RVT_{p,q}^{m}$은 M1-문자열이다.

증명) 가상 토러스 고리 $RVT_{p,q}^{m}$가 매듭이면, 가우스 문자열 $G(RVT_{p,q}^{m})$은 다음과 같이 표현된다.

$$\overline{p-1-2m}+\ldots+\overline{2}+\overline{1}+\sum_{i=1}^{p-2m-1}a_i+$$

① $r=|\{a_i\}|$

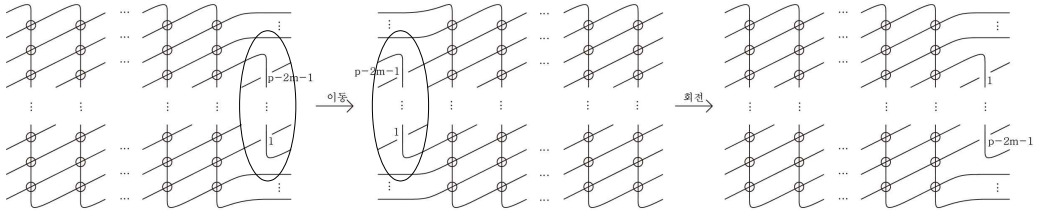
$r=|\{a_i\}|=p-1-2m$

② $a_i\neq a_j,\ \forall i\neq j$, ③ $1\leq a_i\leq r,\ \forall 1\leq i\leq r$

가우스 문자열의 $\sum_{i=1}^{p-2m-1}a_i+$ 부분에 1+부터 r+까지 각각 한 번씩 나와야 하므로 성립한다.

④ $a_i+a_{r-i+1}=r+1,\ \forall 1\leq i\leq r$

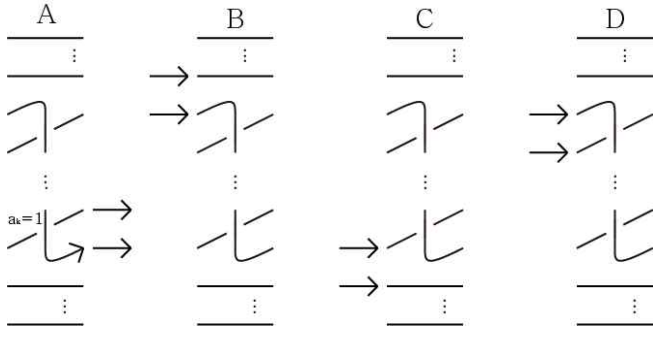
[그림 24]와 같은 과정을 거치면 $G(RVT_{p,q}^{m})$가 대칭적인 구조를 가지는 것을 알 수 있다. 따라서 $a_i=r+1-a_{r-i}$가 성립함을 알 수 있다.



[그림 24] $a_i + a_{r-i+1} = r+1, \ \forall 1 \le i \le r$

⑤ $a_{k+i} = a_i + 1, \ \forall 1 \le i \le r-k$, where $a_k = 1$

[그림 25]의 A에서 볼 수 있듯이 일반적으로 a_{k+i}와 a_i는 한 줄 차이로 나타나므로 위 조건을 만족한다. 만약 위 조건이 성립하지 않는다고 가정하면 [그림 25]의 B, C, D 중 하나의 경우가 나타나는 경우일 것이다. 하지만 B, C의 경우는 발생하면 $i > r-k$거나 가상 토러스 고리가 매듭이 되지 않게 되어 조건을 만족하지 않는다. D의 경우는 발생하기 위해서는 $i = r+k$까지 모든 경우를 만족시킨 이후에 도달할 수 있기 때문에 문제가 되지 않는다. 따라서 가정이 모순이 되고 귀류법에 의해 ⑤가 성립한다.



[그림 25] $a_{k+i} = a_i + 1, \ \forall 1 \le i \le r-k$, where $a_k = 1$

$RVT^m_{p,q}$의 $\{a_i\}$가 M-문자열에 속하기 때문에 변환 H를 사용할 수 있다. 변환 H를 사용하면

$$\overline{r-2x} + \overline{r-2x-1} + \ldots + \overline{2} + \overline{1} + \sum_{i=1}^{r-kx}(a_i) + \sum_{i=r-kx+2}^{k-1}(a_i - 1) + \sum_{i=1}^{r-kx}(a_i - 1) + \sum_{i=r-kx+2}^{k-1}(a_i - 2) + \ldots + \sum_{i=1}^{r-kx}(a_i - (x-1)) + \sum_{i=r-kx+2}^{k-1}(a_i - x) + \sum_{i=1}^{r-kx}(a_i - x) +$$

의 가우스 문자열이 되고, 각 기호에 x를 더하면

$$\overline{r-x}+\overline{r-x-1}+\ldots+\overline{x+2}+\overline{x+1}+\sum_{i=1}^{r-kx}(a_i+x)+\sum_{i=r-kx+2}^{k-1}(a_i+x-1)+$$
$$\sum_{i=1}^{r-kx}(a_i+x-1)+\sum_{i=r-kx+2}^{k-1}(a_i+x-2)+\ldots+\sum_{i=1}^{r-kx}(a_i+1))+\sum_{i=r-kx+2}^{k-1}(a_i)+\sum_{i=1}^{r-kx}(a_i)+$$

과 같은 가우스 문자열이 된다. 위 문자열을 변환 H를 사용하기 전의 가우스 문자열

$$\overline{r}+\ldots+\overline{2}+\overline{1}+\sum_{i=1}^{k-1}a_i+1+\sum_{i=1}^{k-1}(a_i+1)+2+\ldots+x+\sum_{i=1}^{r-kx}(a_i+x)+$$

에 M-문자열의 조건 ④를 바탕으로 $r,\cdots,r+1-x$을 추가한

$$\overline{r}+\overline{r-1}+\ldots+\overline{2}+\overline{1}+\sum_{i=1}^{r-kx}(a_i)+(r+1-x)+\sum_{i=r-kx+2}^{k-1}(a_i)+1+\sum_{i=1}^{r-kx}(a_i+1)+(r-x)+$$
$$\sum_{i=r-kx+2}^{k-1}(a_i+1)+2+\cdots+\sum_{i=1}^{r-kx}(a_i+1))+r+\sum_{i=r-kx+2}^{k-1}(a_i)+x+\sum_{i=1}^{r-kx}(a_i+x)+$$

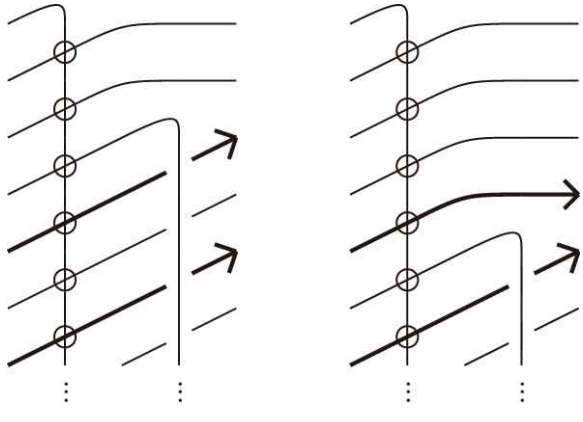
과 비교하면 다음과 같은 차이를 가진다.

① $1,\cdots,x,(r+1-x),\cdots,r$의 기호를 가지고 있지 않고, 대응되는 매듭에서 교차점의 개수가 $2x$개 적다.

② 기존의 $1,\cdots,x,(r+1-x),\cdots,r$를 지나는 부분에서 각 기호를 지나지 않고 값이 1씩 감소한다.

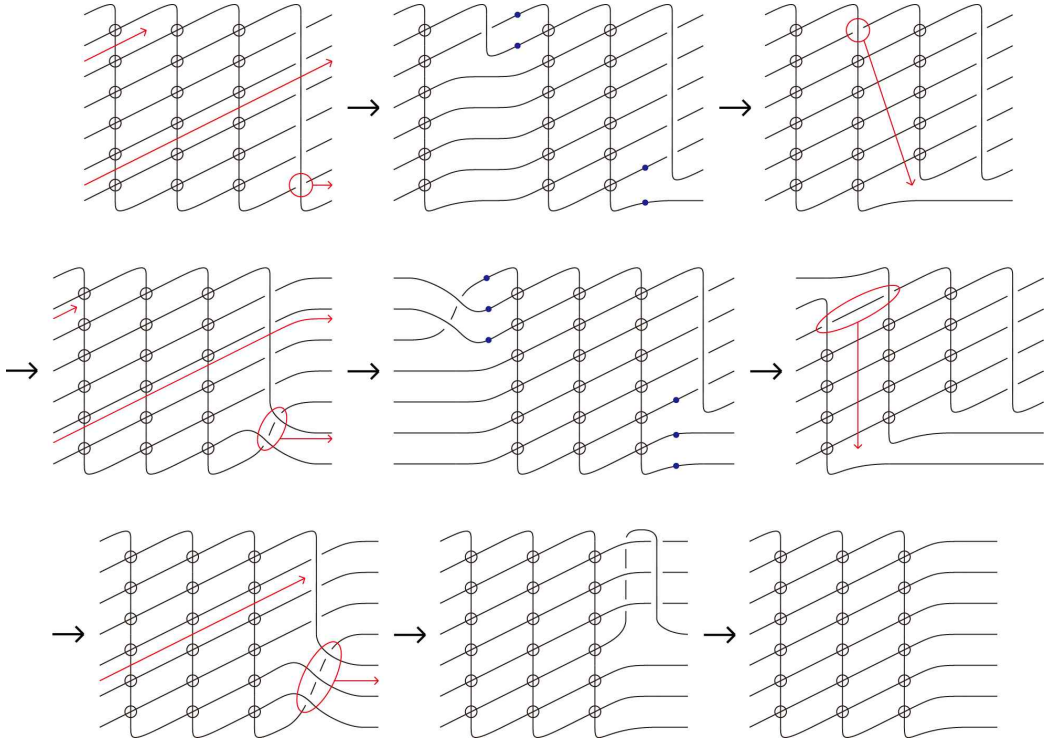
③ 기존의 $x+1,\cdots,r-x$를 지나는 부분은 기존과 같이 지나간다.

[그림 26]를 바탕으로 $RVT_{p,q}^{m}$ 에 변환 H를 사용하면 $RVT_{p,q}^{m+x}$가 되는 것을 알 수 있다.

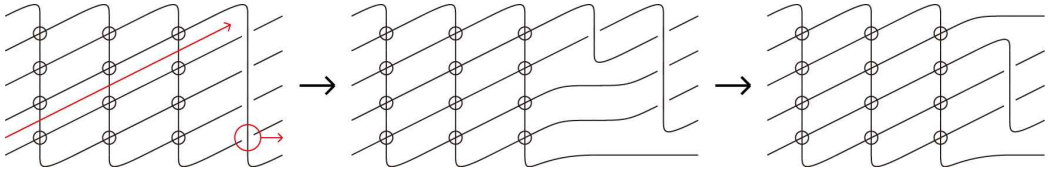


[그림 26] $RVT_{p,q}^{m}$와 $RVT_{p,q}^{m+x}$의 차이

아래 두 개의 예시는 $RVT_{p,q}^{m}$ 에 실제로 변환 L과 P를 이용하여 변환 H를 시행하는 모습을 그린 것이다. 다만, 표현의 편의를 위해 변환 H를 이루는 변환 L과 변환 P가 적용되는 순서는 정리 3.10의 증명에서와는 조금 다르게 하였다. [그림 27]은 $x=3$인 경우, [그림 28]은 $x=1$인 경우이다.



[그림 27] $RVT_{7,4}^{0}$에 대한 변환 H의 과정



[그림 28] $RVT_{5,4}^{0}$에 대한 변환 H의 과정

따름정리 3.12. 가상 토러스 매듭 $VT_{p,q}^{q-1}$에 변환 H를 유한 번 시행하여 M1-문자열이 되게 하거나, 엇갈림이 없는 가우스 문자열이 되게 할 수 있다.

증명) 가상 토러스 매듭 $VT_{p,q}^{q-1}$에 유한 번의 변환 H를 시행한 결과물은 항상 $G(RVT_{p,q}^{m})$의 형태이다. 여기서 $a_1 \neq 1$이면, $G(RVT_{p,q}^{m})$가 M-문자열이므로 변환 H를 또 시행해줄 수 있다. 이렇게 $a_1 = 1$이 되거나, 모든 엇갈림이 사라질 때까지 변환 H를 거듭 시행해줄 수 있다. 또한, 변환 H를 시행해 줄 때마다 최소 2개의 엇갈림은 사라지게 된다. $VT_{p,q}^{q-1}$에 있는 엇갈림의 개수는 유한하므로, 유한 번의 변환 H를 통해 M1-문자열이 되게 하거나($a_1 = 1$이 된 경우), 엇갈림이 없는 가우스 문자열이 되게 할 수 있다(모든 엇갈림이 사라진 경우). □

정리 3.13. $G(VT_{p,q}^{q-1})$ 에 변환 H를 계속 시행해도 $i(c) = 0$인 엇갈림의 개수는 변하지 않는다. 또한, 변환 H를 이용하여 $G(VT_{p,q}^{q-1})$의 $i(c) \neq 0$인 엇갈림을 모두 소거할 수 있다.

증명) 우선, 엇갈림 t에서 $i(c) = 0$이란 의미는 t와 $\bar{t}$ 사이에 위로 지나가는 선분과 아래로 지나가는 선분의 개수가 같다, 즉 아래로 지나가는 선분들 중 t가 t번째에 있다는 것과 같다.

$$\overline{s-1}+...+\bar{2}+\bar{1}+\sum_{i=1}^{k-1}a_i+1+\sum_{i=1}^{k-1}(a_i+1)+2+\sum_{i=1}^{k-1}(a_i+2)+...+x+\sum_{i=1}^{s-1-kx}(a_i+x)+$$

꼴로 표현되는 가우스 문자열에 대해서 변환 H를 시행해주면 $1, 2, ..., x$와 $s-1, s-2, ..., s-x$, 또 그에 대응되는 위로 지나가는 선분들이 RII로 풀리게 된다. 이 때, $1 \leq t \leq x$에 대해 t는 아래로 지나가는 선분들 중 kt번째에 있게 된다. 하지만, $a_1 \neq 1$이므로 $k \geq 2$이고, 따라서 $kt \neq t$임을 알 수 있다. 또한, N의 특성에 의해 $s-t$는 $s-kt$번째에 있게 되고, 같은 이유로 $p-kt \neq p-t$임을 알 수 있다. 따라서 변환 H를 시행해주어도 $i(c) = 0$인 지점이 풀리지는 않는다.

또한, 변환 H를 구성하는 변환 L과 변환 P는 엇갈림 변환과 확장된 라이데마이스터 변환만을 사용하게 되는데, 정리 2.2와 2.3에 의해 이들은 각 엇갈림들의 $i(c)$ 값을 변화시키지 않는다. 따라서 $i(c) = 0$였던 점이 변환 H를 시행한 후에 $i(c) \neq 0$이 되거나, 그 반대의 경우는 생기지 않는다.

그러면 이제 따름정리 3.12에 따라 $G(VT_{p,q}^{q-1})$가 M1-문자열이 되거나 엇갈림이 없는 문자열이 될 때까지 변환 H를 계속 시행하였을 때를 생각해보자. 만약 M1-문자열이 되었다면, M1-문자열과 M1-수열의 정의에 의해 $a_{1+i} = a_i + 1$이므로 $a_i = i$가 성립하고, 따라서 이 때 남아있는 모든 엇갈림들은 $i(c) = 0$이 된다. 즉, $i(c) \neq 0$인 엇갈림들은 모두 소거된 것이다. 또한 엇갈림이 없는 가우스 문자열이 되었다면, 당연히 $i(c) \neq 0$인 엇갈림들도 모두 소거된 것이다. 따라서 항상 변환 H를 이용하여 $G(VT_{p,q}^{q-1})$의 $i(c) \neq 0$인 엇갈림을 모두 소거할 수 있다.

정리 3.14. 가상 토러스 매듭 $VT_{p,q}^{q-1}$의 가상 풀림수 $vu(VT_{p,q}^{q-1})$의 상계는 다음과 같다.

$$vu(VT_{p,q}^{q-1}) \leq \frac{p-1}{2} - \frac{\gcd(p,q-1)-1}{2}$$

증명) 따름정리 3.12에 따라 $G(VT_{p,q}^{q-1})$가 M1-문자열이 되거나 엇갈림이 없는 문자열이 될 때까지 변환 H를 계속 시행하였을 때를 생각해보자. 이 때 d개 엇갈림이 남았다고 하면 총 $(p-1-d)$개의 엇갈림이 변환 H로 풀린 것이다. 이 과정에서 정리 3.10에 의해 $\left(\frac{p-1}{2} - \frac{d}{2}\right)$ 개의 엇갈림 변환이 사용되었다. 그런데, 이 과정 후 가우스 문자열은

$$\overline{d} + \overline{d-1} + \ldots + \overline{2} + \overline{1} + 1 + 2 + \ldots + (d-1) + d +$$

과 같이 표현되게 되며($d=0$인 경우에는 빈 문자열), 이 매듭은 $(d+1,1)$-토러스 매듭이므로 풀린매듭과 동치이다.

정리 3.2와 정리 3.13에 의해

$$d = \gcd(p,q-1)-1$$

가 되므로, 가상 토러스 매듭 $VT_{p,q}^{q-1}$의 가상 풀림수 $vu(VT_{p,q}^{q-1})$의 상계는

$$vu(VT_{p,q}^{q-1}) \leq \frac{p-1}{2} - \frac{\gcd(p,q-1)-1}{2}$$

로 주어진다. □

정리 3.15. 가상 토러스 매듭 $VT_{p,q}^{q-1}$의 가상 풀림수 $vu(VT_{p,q}^{q-1})$는 다음과 같다.

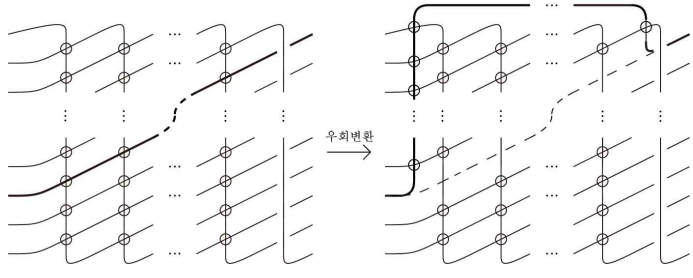
$$vu(VT_{p,q}^{q-1}) = \frac{p-1}{2} - \frac{\gcd(p,q-1)-1}{2}$$

증명) 정리 3.3과 정리 3.14에 의해 성립한다. □

3. 가상화 풀림수 $vvu(VT_{p,q}^{q-1})$의 상계

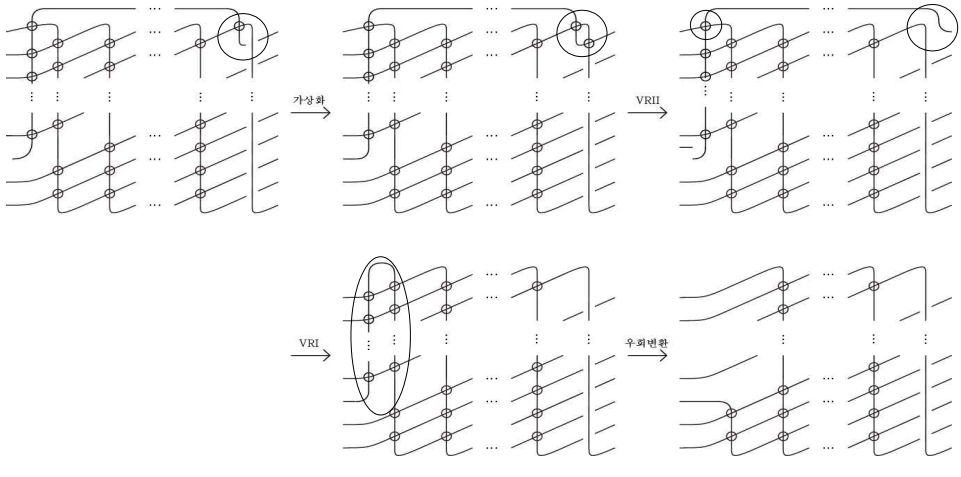
이 단원에서 매듭 K는 K의 땋임과 혼용해서 쓰인다. 또한, 땋임 A와 B, 그리고 자연수 k에 대해 AB는 땋임 A 오른쪽에 땋임 B를 붙인 땋임을 의미하며, kA는 땋임 A를 k개 붙인 땋임을 의미한다. $VT_{1,0}^{-1}$는 풀린매듭을 의미한다.

정의 3.16. $VT_{p,q}^{q-1}$에서 가장 오른쪽 상단에 있는 엇갈림을 지나는 수평 방향의 가닥은 $p>q$이므로 오른쪽에서 왼쪽으로 갈 때 단조감소한다. 따라서 왼쪽 끝에서 이 가닥은 밑에서 $p-q$번째 가닥이 된다. 이제 이 가닥의 모든 가상 엇갈림을 우회 변환을 사용해 가장 왼쪽과 위쪽으로 그림과 같이 모두 옮긴다. 이를 변환 $D1$이라고 하자.



[그림 29] 변환 $D1$

정의 3.17. 변환 $D2$는 $VT_{p,q}^{q-1}$에서 변환 $D1$을 한번 시행한 후 다음과 같은 일련의 시행을 합친 것을 말한다.



[그림 30] 변환 $D2$

보조정리 3.18. $VT_{p,q}^{q-1}$에 변환 $D1$과 변환 $D2$를 순서대로 한 번씩 시행하면

$$VT_{p-q,1}^{1}\, VT_{p-1,q-1}^{q-2}$$

가 된다.

정리 3.19. $VT_{p,q}^{q-1}$에 변환 $D1$과 변환 $D2$를 교대로 각각 $q-1$번 시행하면

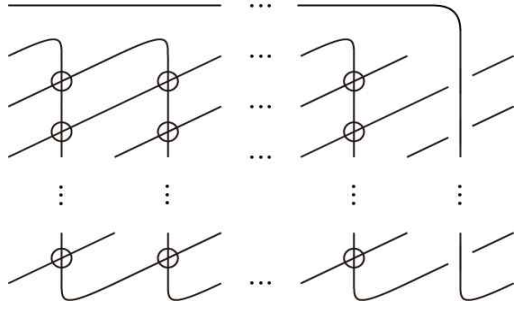
$$\big((q-1)\, VT_{p-q,1}^{1}\big)\, VT_{p-q+1,1}^{0} \sim VT_{p-q,q-1}^{q-1}\, VT_{p-q+1,1}^{0} \sim VT_{p-q,q}^{q-1}$$

가 된다.

증명) $VT_{p,q}^{q-1}$에 변환 $D1$과 변환 $D2$를 한번 시행하면 $VT_{p-q,1}^{1}\, VT_{p-1,q-1}^{q-2}$이 된다. 뒷부분에서는 수평 방향의 가닥과 수직 방향의 가닥이 하나씩 줄어들었고 가상 엇갈림들의 줄도 하나 줄어들었으므로 이 부분에 또다시 변환 $D1$과 변환 $D2$를 시행할 수 있다. 이를 $q-1$회 반복하면

$$\big((q-1)\, VT_{p-q,1}^{1}\, VT_{p-q+1,1}^{0}\big) \sim VT_{p-q,q-1}^{q-1}\, VT_{p-q+1,1}^{0}$$

이 된다. 여기서 $VT_{p-q,q-1}^{q-1}\, VT_{p-q+1,1}^{0}$를 그림으로 나타내면 다음과 같다.



[그림 31] $VT_{p-q,q-1}^{q-1}\, VT_{p-q+1,1}^{0}$

따라서 RI 변환으로 맨 오른쪽 위 엇갈림을 없앨 수 있다. 즉 $VT_{p-q,q}^{q-1}$로 변환된다. □

정의 3.20. $VT_{p,q}^{q-1}$에 변환 $D1$과 변환 $D2$를 교대로 각각 $q-1$번 시행하는 것을 변환 $D3$이라고 한다.

정리 3.21. 가상 토러스 매듭 $VT_{p,q}^{q-1}$과 $VT_{p,\,q \bmod p}^{(q \bmod p)-1}$는 동치이다.

증명) 경우 1) $p > q \Rightarrow q = q \bmod p$

경우 2) $p < q$일 때: $q = jp + l\,(j \in \mathbb{N})$이라고 하자. 수평 방향의 가닥이 가장 왼쪽에서부터 수직 방향의 가닥을 p번 지날 때마다 처음 위치로 돌아온다. 따라서 우회 변환을 사용하여 이 부분을 전부 p개의 수평 방향의 가닥으로 바꿀 수 있다. 이를 총 j번 시행할 수 있으므로 전부 시행하고 나면 $l = q \bmod p$개의 수직 방향의 가닥만 남는다. □

정의 3.22. $p < q$일 때 $VT_{p,q}^{q-1}$을 $VT_{p,q \bmod p}^{(q \bmod p)-1}$으로 변환하는 변환열을 변환 $D4$라고 한다.

보조정리 3.23. 가상 토러스 매듭 $VT_{p,q}^{q-1}$에서 $p = jq + l$로 나타나고 $0 \le l < q$, $l \equiv \pm 1 \pmod q$을 만족하면 다음이 성립한다.

$$vvu(VT_{nq+r,q}^{q-1}) \le n(q-1)$$

증명) $VT_{nq+r,q}^{q-1}$에 변환 $D3$을 한번 시행하면 $q-1$번의 가상화를 하게 되고 $VT_{(n-1)q+r,q}^{q-1}$로 변한다. 따라서 $VT_{nq+r,q}^{q-1}$에는 변환 $D3$을 최대 n번 시행하여 $VT_{r,q}^{q-1}$로 바꿀 수 있고, 이 때 가상화는 $n(q-1)$번 하게 된다. $r \equiv \pm 1 \pmod q$이므로 $r = 1$ 또는 $r = q-1$이다. $r = 1$인 경우는 수평 방향의 가닥이 한 줄이므로 풀린 매듭과 같고, $r = q-1$인 경우는 변환 $D4$를 적용하여 $VT_{q-1,1}^{0}$으로 바꿀 수 있는데, 이것은 $(q-1,1)$-토러스 매듭이므로 풀린매듭이다. □

따름정리 3.24. 가상 토러스 매듭 $VT_{p,q}^{q-1}$에서 $p = jq + l$로 나타나고 $0 \le l < q$, $l \neq \pm 1 \pmod q$을 만족하면 다음이 성립한다.

$$vvu(VT_{nq+r,q}^{q-1}) \le n(q-1) + vvu(VT_{r,q \bmod r}^{(q \bmod r)-1})$$

증명) $VT_{nq+r,q}^{q-1}$에 n번의 변환 $D3$과 한 번의 변환 $D4$를 하는 변환열을 생각해보자. 이 변환열을 한 번 시행한 후에는 $VT_{nq+r,q}^{q-1}$가 $VT_{r,q \bmod r}^{(q \bmod r)-1}$로 바뀐다. 이 때 보조정리 3.23의 경우를 제외한 경우에는 $VT_{r,q \bmod r}^{(q \bmod r)-1}$이 풀린매듭이 아니므로 여기에 또다시 n번의 변환 $D1$과 한 번의 변환 $D3$을 시행하는 변환열을 시행할 수 있다. 따라서 이 경우에는 처음에 한 $n(q-1)$번의 가상화에 $VT_{r,q \bmod r}^{(q \bmod r)-1}$를 풀린매듭으로 만들기 위해 필요한 가상화의 수를 더해줘야 가상화 풀림수가 나오게 된다. □

정리 3.25. 가상 토러스 매듭 $VT_{p,q}^{q-1}$에 대해 다음이 성립한다.

① $vvu(VT_{1,0}^{-1}) = 0$

② $vvu(VT_{n,1}^{0}) = 0\,(n \in \mathbb{N})$

③ $vvu(VT_{p,q}^{q-1}) \le (q-1)\left[\frac{p}{q}\right] + vvu(VT_{p \bmod q, q \bmod (p \bmod q)}^{(q \bmod (p \bmod q))-1})$

증명) 보조정리 3.23에 따라 i)와 ii)는 자명하다. $VT_{p,q}^{q-1} = VT_{nq+r,q}^{q-1}$일 때 $n = \left[\frac{p}{q}\right]$이고 $r = p \bmod q$다. 따라서 $VT_{p,q}^{q-1}$에 $\left[\frac{p}{q}\right]$번의 변환 $D3$과 변환 $D4$를 시행하여 $VT_{p \bmod q, q \bmod (p \bmod q)}^{(q \bmod (p \bmod q))-1}$로 만들 수 있다. 따라서 따름정리 3.24에 의해

$$vvu(VT_{p,q}^{q-1}) \le (q-1)\left[\frac{p}{q}\right] + vvu(VT_{p \bmod q, q \bmod (p \bmod q)}^{(q \bmod (p \bmod q))-1})$$

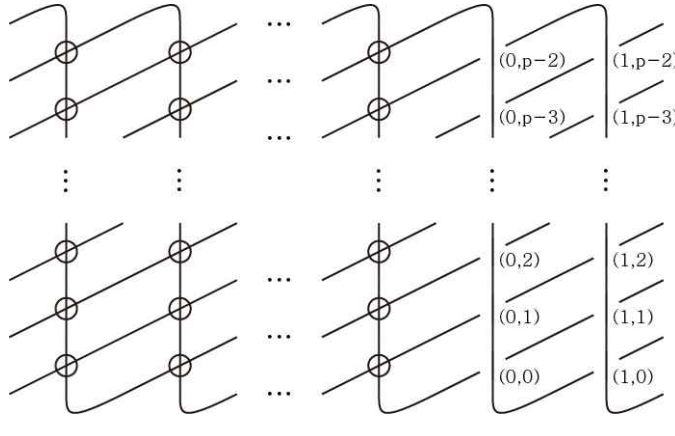
로 나타낼 수 있다. □

정리 3.26. 가상 토러스 매듭 $VT_{p,q}^{q-1}$의 가상화 풀림수 $vvu(VT_{p,q}^{q-1})$의 상계를 정리 3.25만을 이용하여 계산해낼 수 있다.

증명) $vvu(VT_{p,q}^{q-1})$에 대해 정리 3.25의 ③을 한 번 적용해주면 $vvu(VT_{p \bmod q, q}^{q-1})$에 관한 식으로 변하고, 두 번째 단계를 시행하면 $vvu(VT_{p \bmod q, q \bmod (p \bmod q)}^{(q \bmod (p \bmod q))-1})$에 관한 식으로 변한다. 그런데, 여기서 p와 q는 그 두 수에 대해 유클리드 호제법의 알고리즘을 시행하는 것과 같이 변한다. $\gcd(p,q) = 1$이므로 유클리드 호제법에서 p와 q 둘 중 하나는 반드시 1이 된다. 따라서 $vvu(VT_{p,q}^{q-1})$가 결국은 $vvu(VT_{1,0}^{-1}) = 0$ 또는 $vvu(VT_{n,1}^{0}) = 0\,(n \in \mathbb{N})$에 관한 식이 되는데, 이 식은 정리 3.25의 ①과 ②에 의해 계산될 수 있다.

4. 가상 토러스 매듭 $VT_{p,q}^{q-2}$의 가상 풀림수 $vu(VT_{p,q}^{q-2})$의 하계

$VT_{p,q}^{q-2}$의 일반적인 형태는 다음 그림과 같다.



[그림 32] $VT_{p,q}^{q-2}$

정리 3.27. 가상 토러스 매듭 $VT_{p,q}^{q-2}$에서 $kq \equiv -1 \pmod p$인 최소의 양의 정수를 k라 하면,

$$|A_0| = 2 \cdot \left(\left[\frac{\min(k, \frac{p-1}{2}) \cdot \gcd(q-2, p)}{p} \right] + \left[\frac{\min(p-1-k, \frac{p-1}{2}) \cdot \gcd(q-2, p)}{p} \right] \right)$$

이다.

증명) $kq \equiv -1 \pmod p$가 되게 하는 최소의 양의 정수를 k라 하자. 그러면, $VT_{p,q}^{q-2}$의 가상 매듭 다이어그램의 가우스 문자열은 다음과 같다.

$$\overline{(0,p-2)} + \ldots \overline{(0,1)} + \overline{(0,0)} + \sum_{i=1}^{k} ((1,((i-1)q) \bmod p) + (0,(iq-1) \bmod p) +)$$
$$\overline{(1,p-2)} + \ldots \overline{(1,1)} + \overline{(1,0)} + \sum_{i=1}^{p-1-k} ((0,((i+k)q-1) \bmod p) + (1,(i+k)q \bmod p) +)$$

보조정리 3.1에 의해 한 엇갈림의 위로 지나가는 선분과 아래로 지나가는 선분 사이에 있는 위로 지나가는 선분의 개수와 아래로 지나가는 선분의 개수가 같으면 그 엇갈림의 $i(c) = 0$이다. 엇갈림들을 다음 네 가지 경우로 나누어 살펴보자.

경우 1) $1 \le i \le k$일 때 $(0, (iq-1) \bmod p)$과 $\overline{(0, (iq-1) \bmod p)}$

이 경우에는 $\overline{(0, (iq-1) \bmod p)}$보다는 뒤에 있고 $(0, (iq-1) \bmod p)$보다는 앞에 있는 엇갈림들을 고려할 것이다. 위로 지나가는 선분이 $(iq-1) \bmod p$개만큼 있고, 아래로 지나가는 선분이 $2i-1$개만큼 있다. 보조정리 3.1에 의해

$$(iq-1) \bmod p = 2i-1$$

이 성립할 때 그에 대응되는 엇갈림 c에서 $i(c) = 0$이 된다. 위로 지나가는 선분의 개수가 $p-2$보다는 적거나 같을 것이므로 $2i-1 \le p-2$, $i \le \frac{p-1}{2}$도 성립해야 한다. 따라서 $1 \le i \le \min(k, \frac{p-1}{2})$이 된다. 그러면 $1 \le 2i-1 \le p-1$이기 때문에

$$(iq-1) \bmod p = 2i-1$$
$$\Leftrightarrow i(q-2) \equiv 0 \pmod p$$
$$\Leftrightarrow \text{어떤 자연수 } n\text{에 대해 } i = \frac{np}{q-2}$$

가 된다. 따라서 이 식을 만족하는 $1 \le i \le \min(k, \frac{p-1}{2})$ 범위 내에 있는 자연수 i의 개수가 $i(c)=0$인 지점의 개수와 같다.

$a=\gcd(p,q-2)$로 놓으면,

$$i=\frac{np/a}{(q-2)/a}$$

가 되고, $\frac{p}{a}$와 $\frac{q-2}{a}$가 서로소이므로 k가 자연수가 되기 위해서는 어떤 자연수 α에 대해

$$n=\frac{\alpha(q-2)}{a}$$

이다. 따라서

$$i=\frac{\alpha p}{a}$$

이다. 그리고

$$1 \le i \le \min(k, \frac{p-1}{2})$$

이므로

$$1 \le \alpha \le \left[\frac{\min(k, \frac{p-1}{2}) \times \gcd(q-2,p)}{p}\right]$$

이다. 또한, i는 α에 의해 유일하게 결정되므로 $i(c)=0$인 엇갈림들의 개수는

$$\left[\frac{\min(k, \frac{p-1}{2}) \times \gcd(q-2,p)}{p}\right]$$

개이다.

경우 2) $1 \le i \le k$일 때 $(1, ((i-1)q) \bmod p)$과 $\overline{(1, ((i-1)q) \bmod p)}$

이 경우에 해당하는 $(1, x)$ 꼴의 아래로 지나가는 선분들을 뒤에서부터 다시 재배치해보면, $\sum_{i=1}^{k}(1,((k-i)q) \bmod p)+$로 표현이 된다. 따라서 $(1, ((k-i)q) \bmod p)$보다는 뒤에 있고 $\overline{(1, ((k-i)q) \bmod p)}$보다는 앞에 있는 엇갈림들을 고려할 것이다. 이 때 $1 \le i \le k$인 조건은 그대로 유지됨을 알 수 있다. 이제 엇갈림들의 개수를 세보면, 위로 지나가는 선분이 $(p-2)-(((k-i)q) \bmod p)$개만큼 있고, 아래로 지나가는 선분이 $2i-1$개만큼 있다. 보조정리 3.1에 의해

$$(p-2)-(((k-i)q) \bmod p) = 2i-1$$

이 성립할 때 그에 해당하는 점에서 $i(c)=0$이 된다. 위로 지나가는 선분의 개수가 $p-2$보다는 적거나 같을 것이므로 $2i-1 \le p-2$, $i \le \frac{p-1}{2}$도 성립해야 한다. 따라서 $1 \le i \le \min(k, \frac{p-1}{2})$가 된다. 그러면 $1 \le 2i-1 \le p-1$이고, $kq \equiv -1 \pmod p$이기 때문에

$$\begin{aligned} &(p-2)-(((k-i)q) \bmod p) = 2i-1 \\ &\Leftrightarrow i(q-2) \equiv 0 \pmod p \\ &\Leftrightarrow \text{어떤 자연수 } n\text{에 대해 } i = \frac{np}{q-2} \end{aligned}$$

가 된다. 따라서 이 식을 만족하는 $1 \le i \le \min(k, \frac{p-1}{2})$ 범위 내에 있는 자연수 i의 개수가 $i(c)=0$인 지점의 개수와 같고, 경우 1)에서와 같은 논증을 하면 이 경우에도 $i(c)=0$인 엇갈림들의 개수는

$$\left[\frac{\min(k, \frac{p-1}{2}) \times \gcd(q-2, p)}{p} \right]$$

개임을 알 수 있다.

경우 3) $1 \le i \le p-1-k$일 때 $(0,((i+k)q-1) \bmod p)$과 $\overline{(0,((i+k)q-1) \bmod p)}$

이 경우에 해당하는 $(0, x)$ 꼴의 아래로 지나가는 선분들을 뒤에서부터 다시 재배치해보면, $\sum_{i=1}^{p-1-k}(0,((p-i)q-1) \bmod p)+$로 표현이 된다. 따라서 $(0, ((p-i)q-1) \bmod p)$보다는 뒤에 있고

$\overline{(0,((p-i)q-1) \bmod p)}$보다는 앞에 있는 엇갈림들을 고려할 것이다. 이 때 $1 \le i \le p-1-k$인 조건은 그대로 유지됨을 알 수 있다. 이제 엇갈림들의 개수를 세보면, 위로 지나가는 선분이 $(p-2)-(((p-i)q-1) \bmod p)$개만큼 있고, 아래로 지나가는 선분이 $2i-1$개만큼 있다. 보조정리 3.1에 의해

$$(p-2)-(((p-i)q-1) \bmod p) = 2i-1$$

이 성립할 때 그에 해당하는 점에서 $i(c)=0$이 된다. 위로 지나가는 선분의 개수가 $p-2$보다는 적거나 같을 것이므로 $2i-1 \le p-2$, $i \le \frac{p-1}{2}$도 성립해야 한다. 따라서 $1 \le i \le \min(k, \frac{p-1}{2})$가 된다. 그러면 $1 \le 2i-1 \le p-1$이기 때문에

$$\begin{aligned} &(p-2)-(((p-i)q-1) \bmod p) = 2i-1 \\ &\Leftrightarrow i(q-2) \equiv 0 \pmod p \\ &\Leftrightarrow \text{어떤 자연수 } n\text{에 대해 } i = \frac{np}{q-2} \end{aligned}$$

가 된다. 따라서 이 식을 만족하는 $1 \le i \le \min(p-1-k, \frac{p-1}{2})$ 범위 내에 있는 자연수 i의 개수가 $i(c)=0$인 지점의 개수와 같고, 경우 1)에서와 비슷한 논증을 하면 이 경우에도 $i(c)=0$인 엇갈림들의 개수는

$$\left\lfloor \frac{\min(p-1-k, \frac{p-1}{2}) \times \gcd(q-2,p)}{p} \right\rfloor$$

개임을 알 수 있다.

경우 4) $1 \le i \le p-1-k$일 때 $(1,((i+k)q) \bmod p)$과 $\overline{(1,((i+k)q) \bmod p)}$

이 경우에는 $\overline{(1,((i+k)q) \bmod p)}$보다는 뒤에 있고 $(1,((i+k)q) \bmod p)$보다는 앞에 있는 엇갈림들을 고려할 것이다. 위로 지나가는 선분이 $((i+k)q) \bmod p$개만큼 있고, 아래로 지나가는 선분이 $2i-1$개만큼 있다. 보조정리 3.1에 의해

$$((i+k)q) \bmod p = 2i-1$$

이 성립할 때 그에 대응되는 엇갈림 c에서 $i(c)=0$이 된다. 위로 지나가는 선분의 개수가 $p-2$보다는 적거나 같을 것이므로 $2i-1 \le p-2$, $i \le \frac{p-1}{2}$도 성립해야 한다. 따라서 $1 \le i \le \min(p-1-k, \frac{p-1}{2})$이 된다. 그러면 $1 \le 2i-1 \le p-1$이고, $kq \equiv -1 \pmod p$이기 때문에

$$((i+k)q) \bmod p = 2i-1$$
$$\Leftrightarrow i(q-2) \equiv 0 \pmod p$$
$$\Leftrightarrow \text{어떤 자연수 } n\text{에 대해 } i = \frac{np}{q-2}$$

가 된다. 따라서 이 식을 만족하는 $1 \le i \le \min(p-1-k, \frac{p-1}{2})$ 범위 내에 있는 자연수 i의 개수가 $i(c)=0$인 지점의 개수와 같고, 경우 1)에서와 비슷한 논증을 하면 이 경우에도 $i(c)=0$인 엇갈림들의 개수는

$$\left[\frac{\min(p-1-k, \frac{p-1}{2}) \times \gcd(q-2,p)}{p}\right]$$

개임을 알 수 있다.

네 가지 경우들을 모두 합치면, $i(c)=0$인 지점의 개수는

$$2 \cdot ([\frac{\min(k, \frac{p-1}{2}) \times \gcd(q-2,p)}{p}] + [\frac{\min(p-1-k, \frac{p-1}{2}) \times \gcd(q-2,p)}{p}])$$

개가 된다. □

정리 3.28. 가상 토러스 매듭 $VT_{p,q}^{q-2}$에서 $kq \equiv -1 \pmod p$가 되게 하는 최소의 양의 정수를 k라 하면 다음이 성립한다.

$$vu(VT_{p,q}^{q-2}) \ge (p-1) - ([\frac{\min(k, \frac{p-1}{2}) \times \gcd(q-2,p)}{p}] + [\frac{\min(p-1-k, \frac{p-1}{2}) \times \gcd(q-2,p)}{p}])$$

증명) 정리 2.8과 정리 3.27에 의해 성립한다. □

따름정리 3.29. 가상 토러스 매듭 $VT_{p,q}^{q-2}$에서 $q=p-1$인 경우 가상 풀림수 $vu(VT_{p,p-1}^{p-3})$의 하계는 다음과 같이 주어진다.

$$vu(VT_{p,p-1}^{p-3}) \geq \begin{cases} p-2 & (3|p \text{일 때}) \\ p-1 & (\text{그 외의 경우}) \end{cases}$$

증명) $q=p-1$이므로 $k=1$이다. p가 3의 배수일 때에는 $\gcd(q-2,p)=3$이고

$$\begin{aligned} vu(VT_{p,q}^{q-2}) &\geq (p-1)-([\frac{\min(1,\frac{p-1}{2})\times 3}{p}]+[\frac{\min(p-1,\frac{p-1}{2})\times 3}{p}]) \\ &=(p-1)-(0+1)=p-2 \end{aligned}$$

이다. p가 3의 배수가 아니면, $\gcd(q-2,p)=1$이고

$$\begin{aligned} vu(VT_{p,q}^{q-2}) &\geq (p-1)-([\frac{\min(1,\frac{p-1}{2})\times 1}{p}]+[\frac{\min(p-1,\frac{p-1}{2})\times 1}{p}]) \\ &=(p-1)-(0+0)=p-1 \end{aligned}$$

이다. □

IV. 참고문헌

[1] Masaharu Ishikawa and Hirokazu Yanagi. Virtual unknotting numbers of certain virtual torus knots. *arXiv preprint math/1701.03999*, 2017.

[2] Heather A. Dye, *An invitation to knot theory: virtual and classical*. CRC Press, Illinois, 2016.

[3] Louis H. Kauffman and Sofia Lambropoulou, Virtual braids. *arXiv preprint math/0407349*, 2004.

A. 부록 1: 가우스 문자열 생성, P-불변량 계산 알고리즘

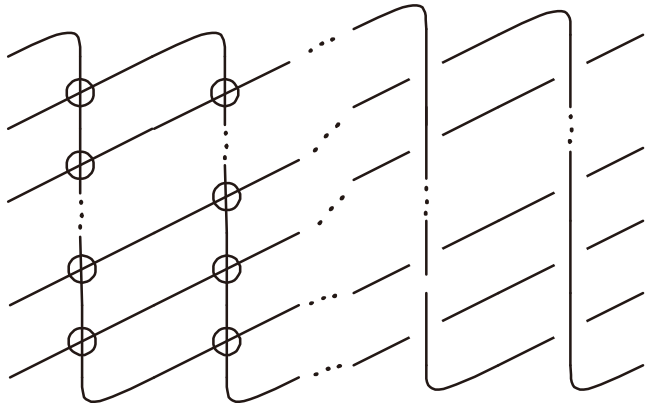
1. 가우스 문자열 생성 알고리즘

논문에서 가상 토러스 매듭의 가상 풀림수의 하계를 계산할 때 P-불변량을 이용하였다. 일반적인 경우라면 매듭의 가우스 문자열은 규칙성이 거의 없기 때문에 가우스 문자열을 계산하기 힘들지만, 가상 토러스 매듭은 대칭성이 있어 가우스 문자열에 규칙성을 찾을 수 있기 때문에 P-불변량을 계산하는 알고리즘을 비교적 쉽게 짤 수 있다. 그래서 우리는 가상 토러스 매듭의 하계의 규칙성을 찾기 위해 몇 가지 알고리즘을 고안하였다. 먼저 가상 토러스 매듭의 엇갈림의 위치를 나타내는 행렬을 정의한다.

정의 A.1. 집합 $C(VT^n_{p,q})$를 다음과 같이 정의한다.

$$C(VT^n_{p,q}) := \{c | c\text{는 } VT^n_{p,q}\text{의 엇갈림}\}$$

정의 A.2. 가상 토러스 매듭 다이어그램 $VT^n_{p,q}$이 있을 때, 엇갈림에 다음과 같이 순서대로 기호를 붙여준 뒤 행렬 $M_{C(VT^n_{p,q})}$를 다음과 같이 정의한다.



[그림 A-1] $VT^n_{p,q}$의 엇갈림의 기호

$$M_{C(VT^n_{p,q})} \in \mathbf{M}_{(p-1)\times q}\big(C(VT^n_{p,q})\big), M_{C(VT^n_{p,q})} = \begin{bmatrix} C_{00} & C_{01} & \cdots & C_{0q-2} & C_{0q-1} \\ C_{10} & C_{11} & \cdots & C_{1q-2} & C_{1q-1} \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ C_{p-30} & C_{p-31} & \cdots & C_{p-3q-2} & C_{p-3q-1} \\ C_{p-20} & C_{p-21} & \cdots & C_{p-2q-2} & C_{p-2q-1} \end{bmatrix}$$

정의 A.3. 가상 토러스 매듭 다이어그램 $VT^n_{p,q}$ 에서 가로선이 p줄, 세로선이 q줄일 때, 각 선에 0에서부터 시작하여 각각 p-1, q-1까지 번호를 매겨준 뒤, 다음의 정수형 변수 r과 c를 정의한다.

$$r := \text{현재 있는 가로줄}\,(0 \le r \le p-1)$$
$$c := \text{현재 있는 세로줄}\,(0 \le c \le q-1)$$

정의 A.4. 각 원소에 순서가 있으며 순서가 유지되는 집합을 순서집합이라고 한다. 순서집합의 모든 원소는 집합 내에서의 원소의 순서를 나타내는 인덱스를 가진다. 이 때, 인덱스는 0에서 시작하여 순서대로 1씩 증가하는 0 이상의 정수다.

정의 A.5. 가상 토러스 매듭 다이어그램 $VT^n_{p,q}$ 에 대하여 $GW(VT^n_{p,q})$을 $VT^n_{p,q}$ 의 가우스 문자열로 정의한다. 이 때, $GW(VT^n_{p,q})$는 순서집합이다.

어떤 방향 가상 매듭의 가우스 문자열은 시작점에 상관없이 모두 동치이다. 따라서 알고리즘의 시작점을 임의로 정할 수 있는데, 여기서는 편의를 위해 가상 토러스 매듭의 왼쪽 상단으로 정한다. 즉 초기 위치는 0번째 가로줄, 0번째 세로줄이다. 가상 토러스 매듭의 가우스 문자열을 구하는 알고리즘은 다음과 같다. 참고로 방향은 왼쪽 위에서 오른쪽 아래 방향이다. 따라서 모든 엇갈림의 부호는 +다.

① r과 c를 각각 0으로 초기화한다. 즉, 매듭의 왼쪽 상단에서 시작한다. 그리고 알고리즘의 시행 횟수를 세는 카운터 cnt를 정의하고 0으로 초기화한다. 또한 공순서집합 $GW(VT^n_{p,q})$를 정의한다.

② 먼저 첫 번째 세로줄에서 시작한다. 따라서 c를 0으로 한다.

③ c가 q 미만인지 확인한다. 만약 아니라면 cnt를 1 증가시킨 뒤 cnt가 p 미만이면 ②로 돌아가고, 아니라면 알고리즘을 종료한다. c가 q 미만이면 ④로 이동한다.

④ r이 0인지 확인한다. r이 0이면 ⑤로 이동하고, 아니라면 ⑦로 이동한다.

⑤ r이 0인 경우 방향대로 진행할 때 맨 윗줄에서 아랫줄로 이동하게 된다. 따라서 그 사이의 세로선에 있는 모든 엇갈림을 위로 지나게 되므로 이를 가우스 문자열에 더해줘야

한다. 이 때, c가 n 이상인지 확인한다. 만약 그렇다면 c번째 세로줄의 엇갈림은 모두 가우스 문자열에 나타나므로 $GW(VT^n_{p,q})$에 $\overline{C_{0c}}$+부터 $\overline{C_{p-2c}}$+까지 순서대로 더한다.

⑥ r을 p-1로 설정하여 맨 아랫줄로 이동하고, c에 1을 더하여 다음 세로줄로 이동한다. 그리고 ③으로 돌아간다.

⑦ r이 0이 아니라면 다음 세로줄로 이동하면서 엇갈림을 아래로 지나게 된다. 따라서 이를 가우스 문자열에 더해줘야 한다. 이 때, c가 n 이상인지 확인한다. 만약 그렇다면 그 엇갈림은 가우스 문자열에 나타나므로 $GW(VT^n_{p,q})$에 C_{r-1c}+를 더한다.

⑧ r에 1을 빼서 윗줄로 이동하고, c에 1을 더하여 다음 세로줄로 이동한다. 그리고 ③으로 돌아간다.

위의 알고리즘에 의하면 ②에서 시작하는 루프를 총 p번 반복하게 된다. 따라서 p와 q가 서로소여서 $VT^n_{p,q}$이 매듭인 경우에는 위 알고리즘으로 모든 엇갈림을 위로 한 번, 아래로 한 번씩 지나가게 되어 유효한 가우스 문자열이 만들어진다. 만약 p와 q가 서로소가 아니면 같은 문자열이 반복되는 구간이 생기게 된다. 그 외에도 알고리즘적인 오류로 인해 유효하지 않은 가우스 문자열이 만들어질 수 있다. 이를 방지하기 위해 가우스 문자열이 유효한지 확인하여야 한다.

정의 A.6. 가상 토러스 매듭 $VT^n_{p,q}$의 가우스 문자열 GW가 있을 때, $S(GW)$를 정렬된 가우스 문자열로 정의한다. 이 때, 정렬 기준은 GW의 원소에 해당하는 $M_{C(VT^n_{p,q})}$의 성분을 기준으로 정한다. 이 때, 임의의 두 원소 $w_1, w_2 \in GW$가 있을 때, 이 두 원소에 해당하는 $M_{C(VT^n_{p,q})}$의 성분이 서로 같은 경우 w_1과 w_2의 인덱스는 1 차이나도록 한다.

정의 A.7. 가우스 문자열 GW와 정렬된 가우스 문자열 $S(GW)$의 i번째 원소 w_i에 대해 다음의 함수 c, s, p를 정의한다.

$$c(w_i) := w_i\text{에 해당하는 } M_{C(VT^n_{p,q})}\text{의 성분}$$

$$s(w_i) := w_i\text{의 부호}$$

$$p(w_i) := \begin{cases} OVER & (w_i\text{가 엇갈림을 위로 지나가는 경우}) \\ UNDER & (w_i\text{가 엇갈림을 아래로 지나가는 경우}) \end{cases}$$

정리 A.8. 정렬된 가우스 문자열 $S(GW)$에 대해

① $c(w_i) = c(w_{i+1}) = c(w_{i+2})$인 i가 존재하거나

② $c(w_i) = c$를 만족하는 i가 유일하게 존재하는 집합 $\{c(w_i) | w_i \in S(GW)\}$의 원소 c가 존재하거나

③ $s(w_i) \neq s(w_{i+1})$을 만족하는 2의 배수 i가 존재하거나

④ $p(w_i) = p(w_{i+1})$을 만족하는 2의 배수 i가 존재할 경우

가우스 문자열 GW는 유효하지 않다. 그 외의 경우에는 유효하다.

증명)

1. $c(w_i) = c(w_{i+1}) = c(w_{i+2})$가 성립한다면 가우스 문자열 상에서 매듭의 한 엇갈림을 최소한 세 번 지나간다는 의미이다. 하지만 방향 가상 매듭에서 하나의 엇갈림은 항상 위로 한 번, 아래로 한 번, 총 두 번 지나가게 되므로 이러한 상황은 존재하지 않는다. 따라서 위의 상황이 성립할 경우 잘못된 가우스 문자열이다.

2. $c(w_i) = c$를 만족하는 i가 하나밖에 없는 c가 존재한다는 것은 그 엇갈림을 한 번밖에 지나지 않는다는 의미이다. 따라서 위에서 말했듯이 그런 상황은 존재할 수 없다.

3. $S(GW)$이 ①번과 ②번 테스트를 통과했다고 가정하였을 때, 모든 원소는 같은 엇갈림을 가진 원소끼리 두 개씩 모여 있을 것이다. 따라서 인덱스가 2의 배수인 원소와 그 다음 원소의 엇갈림은 같다. 따라서 두 원소의 부호는 같아야 한다. 그렇기 때문에 만약 두 원소의 부호가 다르다면 그 가우스 문자열은 유효하지 않다.

4. 3번 상황에서 인접한 두 원소쌍이 둘 다 엇갈림을 위로 지나가거나 아래로 지나가는 일은 존재할 수 없다. 따라서 이러한 경우에도 가우스 문자열은 유효하지 않다.

① 번 테스트에서 같은 엇갈림에 해당하는 원소가 4개 이상인 경우에도 $c(w_i) = c(w_{i+1}) = c(w_{i+2})$가 성립하므로 이는 모두 ①을 만족하는 경우이고, ②번 테스트에서는 어떤 엇갈림에 해당하는 원소가 1개인 경우를 찾아낸다. 또 ③번 테스트에서는 같은 엇갈림의 부호가 다른 경우를 찾아내며, ④번 테스트에서는 같은 엇갈림에서 두 번 모두 위로, 또는 아래로 지나가는 경우를 찾아낸다. 따라서 위의 네 가지 경우를 제외하고 나면 같은 엇갈림에 해당하는 원소가 모두 2개이고 같은 엇갈림의 부호가 모두 같으며, 모든 엇갈림을 위로 한 번, 아래로 한 번 지나가는 경우만 남는다. 이러한 가우스 문자열은 유효하다. □

2. P-불변량 계산 알고리즘

방향 가상 매듭 다이어그램의 가우스 문자열을 알면 가우스 다이어그램을 그릴 수 있으므로 매듭의 P-불변량을 계산할 수 있다. P-불변량의 계산에서 핵심적인 부분은 화살표를 중심으로 가우스 다이어그램의 원호를 두 부분으로 나누는 것이다. 가우스 다이어그램과 동치인 가우스 문자열 상에서 이는 함수 $c(w_i)$의 값이 같은 한 쌍의 가우스 문자열 GW의 원소 사이에 있는 모든 원소와 나머지 원소로 나누는 것으로 생각할 수 있다. 알고리즘은 다음과 같다.

① P-불변량에 해당하는 다항식 $P(K)$를 정의하고 0으로 초기화한다. 또, 공순서집합 $\gamma_1(K)$와 $\gamma_2(K)$를 정의한다. 이 둘은 각각 가우스 다이어그램 상에서 원호 γ_1과 γ_2 위에 존재하는 가우스 문자열 $GW(K)$의 원소의 집합이다. 또 공집합 D를 정의한다. 이는 계산한 $GW(K)$의 원소의 집합이다.

② $GW(K)$가 유효한지 확인한다. 유효하지 않으면 알고리즘을 종료한다.

③ 카운터 i를 0으로 초기화한다.

④ $GW(K)$의 원소 w_i에 대해 w_i와 함수 c의 값이 같은 원소가 D에 존재하는지 확인한다. 존재한다면 그 원소는 이미 계산된 것이므로 ⑭로 넘어간다.

⑤ $j > i$이고 $c(w_i) = c(w_j)$인 정수 j를 찾는다. ②번 과정에서 가우스 문자열이 유효하다는 것을 확인했으므로 이러한 정수는 반드시 존재한다.

⑥ $i < k < j$인 모든 $GW(K)$의 원소 w_k를 $\gamma_1(K)$에 넣는다. 그리고 w_i와 w_j를 제외한 나머지 원소는 $\gamma_2(K)$에 넣는다.

⑦ $i(c(w_i))$를 0으로 초기화한다.

⑧ 카운터 k를 0으로 초기화한다.

⑨ $\gamma_1(K)$의 원소 w_k에 대해 $c(w_k) = c(e)$인 원소 e가 $\gamma_2(K)$에 존재하는지 확인한다. 없다면 w_k에 해당하는 화살표가 w_i에 해당하는 화살표를 관통하지 않는 것이므로 P-불변량의 계산에서 제외되다. • 따라서 이러한 경우에는 ⑪로 넘어간다.

⑩ $p(w_k)$가 $OVER$인지 확인한다. $OVER$라면 w_k에 해당하는 화살표가 γ_1에서 출발한다는 뜻이며, $UNDER$라면 γ_1로 들어온다는 의미이다. P-불변량을 계산할 때 $i(c(w_i))$의 절댓값으로 계산하므로 어느 쪽을 뒤집든지 상관없다. 여기서는 $p(w_k)$가 $OVER$인 경우 $i(c(w_i))$에 $s(w_k)$를 더하고, $UNDER$인 경우 $i(c(w_i))$에 $s(w_k)$를 뺀다.

⑪ k에 1을 더하고 k가 $|\gamma_1(K)|$ 미만이라면 ⑨로 돌아가고 아니라면 ⑫로 넘어간다.

⑫ $i(c(w_i))$가 0이 아닌 경우 그 화살표는 P-불변량의 계산에 포함되므로 $P(K)$에 $s(w_i)t^{|i(c(w_i))|}$를 더한다.

⑬ w_i를 D에 넣는다.

⑭ i에 1을 더하고 i가 $|GW(K)|$ 미만이라면 ④로 돌아가고 아니라면 알고리즘을 종료한다.

3. 알고리즘의 구현

자바로 구현한 위의 알고리즘의 구현체는 다음과 같다.

GaussLetter

가우스 문자열의 원소의 구현 클래스이다. 필드에는 엇갈림을 위로 지나가는지 아래로 지나가는지 나타내는 중첩 클래스인 열거형 CROSSING과 엇갈림의 부호를 나타내는 열거형 SIGN, 원소에 해당하는 엇갈림을 나타내는 문자열 변수 notation, 엇갈림이 지나가는 위치를 나타내는 CROSSING형 변수 crossing, 엇갈림의 부호를 나타내는 SIGN형 변수 sign이 선언되어 있다.

GaussLetter: 변수 세 개를 받아서 필드를 초기화하는 생성자다.

toString: Object 클래스의 toString을 오버라이드한 메소드로 원소를 나타내는 문자열을

리턴한다. getXXX: 모두 필드를 반환하는 getter 메소드이다.

equals: Object 클래스의 equals를 오버라이드한 메소드로 파라미터로 받은 GaussLetter 인스턴스와 같은지 확인한다. notation이 같으면 같은 것으로 본다.

compareTo: Comparable〈〉 인터페이스의 구현 메소드로 notation의 compareTo 메소드의 리턴값을 리턴한다. 원소를 notation을 기준으로 사전 순으로 배열하도록 한다.

hashCode: Object 클래스의 hashCode를 오버라이드한 메소드로 인스턴스의 해시코드를 리턴한다.

GaussWord

가우스 문자열의 구현클래스이다. 가우스 문자열을 GaussLetter의 배열로 생각하여 필드에는 GaussLetter의 LinkedList인 word가 선언되어있다.

GaussWord(): 비어있는 가우스 문자열을 만드는 생성자다.

GaussWord(GaussLetter... letters): 파라미터로 전달된 GaussLetter 인스턴스를 순서대로 word에 추가하여 초기화하는 생성자다.

addLetter: 파라미터로 전달된 GaussLetter 인스턴스를 가우스 문자열의 맨 끝에 붙이는 메소드다.

isConsistent: 정리 A.8의 구현 메소드로 가우스 문자열이 유효한지 확인하여 유효하면 true를, 유효하지 않으면 false를 리턴한다.

pInvariant: A.2.의 P-불변량 계산 알고리즘의 구현 메소드다. 가우스 문자열이 유효하지 않으면 Exception을 던지고, 유효하다면 P-불변량인 다항식 인스턴스 Polynomial을 리턴한다.

getUnderBoundOfVirtualUnknottingNumber: P-불변량을 계산하고 이를 통해 가우스 문자열에 해당하는 매듭의 가상 풀림수의 하계를 리턴한다. pInvariant 메소드를 내부에서 호출하므로 Exception을 던진다.

Monomial

단항식의 구현 클래스다. 필드에는 변수를 나타내는 문자열 변수 unknown, 계수를 나타내는 정수형 변수 coefficient, 차수를 나타내는 정수형 변수 dimension이 선언되어 있다.

Monomial: 변수 세 개를 받아서 필드를 초기화하는 생성자다.

equals: 파라미터와 unknown과 dimension이 모두 같으면 true를 리턴한다.

add: 파라미터로 받은 Monomial 인스턴스로 equals를 호출하여 true이면 두 단항식을 더한다. 두 단항식이 같지 않다면 서로 더하면 다항식이 되므로 ArithmeticException을 던진다.

compareTo: 단항식을 변수를 나타내는 문자의 사전순으로 정렬하게 한다. 두 단항식의

변수의 문자가 같다면 차수를 기준으로 내림차순으로 정렬한다. 차수도 같으면 계수를 기준으로 오름차순으로 정렬한다.

Polynomial

다항식의 구현 클래스다. 다항식을 단항식의 합으로 볼 수 있으므로 필드에는 Monomial의 LinkedList인 poly가 선언되어있다.

Polynomial(): 비어있는 다항식을 생성하는 생성자다.

Polynomial(Monomial... terms): 파라미터로 전달된 Monomial 인스턴스를 순서대로 poly에 추가하여 초기화하는 생성자다.

getAbsoluteSumOfCoefficients: 다항식의 계수의 절댓값의 합을 리턴한다. 이 메소드는 GaussWord 클래스의 getUnderBoundOfVirtualUnknottingNumber 메소드에서 호출되어 매듭의 가상 풀림수의 하계를 계산하는데 쓰인다.

add(Polynomial poly): 파라미터로 전달된 다항식을 더하는 메소드다.

add(Monomial mon): 파라미터로 전달된 단항식을 더하는 메소드다.

VirtualTorus

$VT^n_{p,q}$의 P-불변량을 구해 가상 풀림수의 하계를 계산하는 클래스다. 인스턴스의 생성이 필요하지 않기 때문에 abstract로 선언되었다.

underbound: 정적 메소드로 p, q, n을 받아서 $VT^n_{p,q}$의 가우스 문자열을 만든 뒤, P-불변량을 계산하여 가상 풀림수의 하계를 계산한 뒤 p, q, n, 가상 풀림수의 하계의 배열을 리턴한다. 내부에 A.1.의 가우스 문자열을 구하는 알고리즘이 구현되어 있다.

위의 프로그램을 통해 p와 q가 서로소이고 n이 q-1인 경우의 가상풀림수의 하계를 구할 수 있었고, 이를 통해 $n(A_0)$를 알 수 있었다. 그 결과 중 다음은 $p=2(q-1)$인 경우이다.

〈표 A-1〉 $VT^n_{p,q}$ 의 $\frac{n(A_0)}{2}$

p	q	n	$\frac{n(A_0)}{2}$
8	5	4	1.5
12	7	6	2.5
16	9	8	3.5
20	11	10	4.5
24	13	12	5.5
28	15	14	6.5
32	17	16	7.5
36	19	18	8.5
40	21	20	9.5
44	23	22	10.5
48	25	24	11.5
52	27	26	12.5
56	29	28	13.5
60	31	30	14.5
64	33	32	15.5
68	35	34	16.5
452	227	226	112.5
456	229	228	113.5
460	231	230	114.5
464	233	232	115.5
468	235	234	116.5
472	237	236	117.5
476	239	238	118.5
480	241	240	119.5
484	243	242	120.5
488	245	244	121.5
492	247	246	122.5
496	249	248	123.5

위의 표에서 알 수 있는 사실은 $(q-1)-1=n(A_0)=\gcd(p,q-1)-1$라는 것이다. 우리는 이 규칙이 $p=2(q-1)$가 아닌 다른 일반적인 경우에도 적용된다고 추론하였고, 이에 정리 3.2.를 추론할 수 있었다.

ABSTRACT

Research on Unknotting Sequence and Unknotting Number of Certain Virtual Torus Knots

Reseacher : **Subo Lee(sophomore, suboo00@naver.com)**
Jinwuk Pee(sophomore, jwpee@naver.com)
Junyoung Heo(sophomore, joeheomail@naver.com)

Supervisor : **Hun Kim(Korea Science Academy of KAIST, HunKim@kaist.ac.kr)**
Co-Supervisor : **Seungick Lee(Korea Science Academy of KAIST, maick@hanmail.net)**

Abstract

We calculated unknotting number of $VT_{p,q}^{q-1}$, which is a knot created by changing all crossings in (p,q)-torus knot to virtual crossings, except for the last line. Unknotting number means how many crossing changes are needed to make a knot to an unknot. Using the P-invariant, the lower bound of this value was shown to be $(p-\gcd(p,q-1))/2$. Also, by finding out approximate unknotting sequence, the upper bound of this value was shown to be exactly the same with the lower bound under some assumptions. Furthermore, we calculated the upper bound of unknotting number where only virtualization is allowed instead of crossing changes, and also figured out the lower bound of unknotting number of $VT_{p,q}^{q-2}$, a knot created by changing all crossings to virtual crossings, except for the last two lines.