Algorithms for Braid representation from Gauss code

Cho, Myeong-Sang

September 12, 2023

1 가우스 코드란 무엇인가?

$\mathbb{R}^3$ 위의 매듭 혹은 링크 K에 대해, 충분히 좋은 2차원으로의 투영 L을 생각해봅시다.

이때, 충분히 좋다는 것은 L은 자기 자신과 오직 X자의 형태로만 만나도록 되어 있고, 자기 자신과 접하거나 하지는 않는다는 뜻입니다.(수학적으로는 its self-intersection is only transverse double point 정도로 표현할 수 있겠습니다.)

그럼 이때, L이 스스로 만나는 지점들을 교차점(crossing)이라고 하고, 숫자를 배당하겠습니다.

반드시 유념하실 것은, 각 교차점을 지나는 선분들에 대해서는 위 아래가 존재한다는 사실입니다. 매듭과 링크는 3차원 대상이었으니 투영하면 이 정보가 사라지지만, 이것을 어떤식으로든 기억하게 만들 수 있습니다.

L의 형태는 궁극적으로 원들이 X자로 교차하면서 모인 모임으로 볼 수 있는데, 각 원마다 교차점이 아닌 지점에 점을 하나씩 배당합시다. 그리고나서, 이 점이 원을 따라갈 방향을 고정합니다. 시계방향이든 시계반대방향이든, 점이 원 전체를 둘러보고 오도록 할 수 있습니다.

이때, 점이 원을 따라 가면서 만나는 교차점들을, 위-아래 정보와 함께 차례대로 기록합니다.

그러면 우리는 이렇게 만들어진 수열을 원의 갯수만큼 갖게 됩니다. 이것을 **가우스 코드**라고 합니다.

세잎 매듭의 특정 투영의 경우는, 가우스 코드로

$$1^+2^-3^+1^-2^+3^- \tag{1}$$

를 가집니다.

2 가우스 코드가 실제 매듭을 나타냄을 보이기

우리는 가우스 코드를 추상화할 수 있습니다. "가우스 코드"가 될만한 수열들의 모임을 생각하면

1. 같은 숫자는 반드시 어디선가 2번만 등장한다.
2. 같은 숫자가 등장하되, 부호는 달라야 한다.

같은 규칙을 지켜야 한다는 걸 알 수 있습니다.

그러나 이러한 "유사 가우스 코드"들이 전부 어떤 매듭의 투영에 대응될까요?

이 문제는 많은 사람들이 이미 연구한 주제입니다. 답은 "아니다"이고,

$$1^+2^-1^-2^+ \tag{2}$$

같은 예제들이 있습니다.

그러면 우리는 어떻게 주어진 유사 가우스 코드가 실제로 가우스 코드임을 보일 수 있을까요?

다음의 알고리즘을 따라하면 됩니다. 여기서 설명하는 알고리즘은, 매듭 이론과 퍙음 이론의 강력한 연결고리를 활용합니다. 증명 역시 첨부할테니 참고하세요.

2.1 버팀줄(strand) 찾기

1. 투영 L의 각 원마다 교차점과 교차점 사이의 선분마다 하나의 표시점을 배당합니다.
2. 모든 교차점을 검색해서, 교차점 뒤(이미 원마다 방향이 주어져있음을 가정합니다.)의 두 선분에 표시점이 있는 교차점을 찾으면 이를 지우고, 앞의 두 선분에 표시점을 그려주되, 이미 표시점이 있는 선분에는 표시하지 않고 넘어갑니다.

   이 작업이 끝나면, 해당 교차점의 주변 선분에는 두 개의 교차점 밖에 없습니다.
3. 이 작업을 교차점의 갯수만큼만 반복하면 됩니다. 남는 표시점들을 그대로 두고 "버팀줄"이라고 부르기로 합시다.

Proof. 교차점의 갯수에 대한 수학적 귀납법 : 위의 알고리즘은 반드시 종료된다.
교차점이 0개이거나 하나인 경우를 생각하면 자명하게 참입니다.
이제 N개일 때 성립한다고 가정하면,
$N+1$개일 때, 첫 교차점 근처를 지우고, 선분들끼리 교차하지 않게 다시 연결해주면 N개의 교차점을 갖는, 원래의 투영과는 전혀 다른 새로운 투영을 얻을 수 있습니다.
그런데, 이 과정은 교차점으로 분리된 선분에 위치한 표시점들에 대한 알고리즘을 수행한 것과 본질적으로 같습니다.
N개의 교차점을 갖는 새로운 투영에 대해 알고리즘이 수행된다고 가정했으므로, 우리는 $N+1$개의 투영에 대해서도 이것이 가능하다는 것을 알 수 있습니다.
따라서 수학적 귀납법에 따라 증명됩니다. □

2.2 유사 가우스 코드의 버팀줄 찾기

위의 정리는 실제 매듭의 투영에 대한 것이고, 이를 유사 가우스 코드에 대해서도 동일하게 할 수 있습니다.

1. 가우스 코드의 숫자 사이마다 'X' 표시를 넣습니다.
2. 'X'가 동일한 숫자를 가진 두 교차점 앞에 등장하는 경우를 찾으면, 앞에 위치한 'X'를 지우고 숫자의 뒤에 적습니다.

   이때, 'X'가 두개가 되면 'X'를 하나만 남깁니다.

   (한편, 이 숫자들을 차례대로 모두 기록해둡니다. 버팀줄의 존재와는 별개로 나중에 필요합니다.)
3. 이 작업을 숫자들의 갯수만큼 수행합니다. 그러면 결과는 버팀줄의 위치가 됩니다.

   (위에서 숫자들을 기록해두면, 수열의 모임도 얻을 수 있습니다.)
4. 만약 중간에 이 과정을 수행할 수 없는 경우가 발생한다면, 매듭의 투영이 아니라는 뜻이므로, 매듭이 아니다 라고 얘기할 수 있습니다.

그러나, 버팀줄의 존재성만으로는 여전히 유사 가우스 코드가 실제 가우스 코드인지 알 수는 없습니다. 거의 대부분의 유사 가우스 코드는 버팀줄을 가집니다.(어쩌면 전부. 이를 증명할 수도 있을 것입니다. 하지만 여기서는 생략합니다.) 우리는 한발자국 더 나아가야 합니다.

2.3 더 많은 보조정리들

가장 중요한 보조정리 : 버팀줄을 구할 때 작업을 수행한 교차점의 순서는 버팀줄이 이동할 순서와 같다.

Proof. 먼저, 교차점의 갯수에 대한 수학적 귀납법으로, 실제로 버팀줄이 알고리즘이 시작한 처음 위치로 돌아오겠는가?를 먼저 보입니다.

교차점이 0개거나 1개라면 자명합니다.

N보다 작은 k들에 대해 이것이 성립한다고 가정합시다.

교차점의 갯수가 N+1인 L를 생각하였을 때, 버팀줄을 찾는 알고리즘에서 수행한 것과 동일한 작업을 수행해봅시다. 그러면, 동등하게 교차점의 갯수가 N인 L'에 대해 작업을 수행한 셈입니다.

가정에 따라, 최후에 돌아오는 지점은 처음 지접인데, 이 지점은 사실 L의 지워진 교차점의 앞부분 (혹은 뒷부분)과 같습니다. 어느쪽이든 L의 경우에서도 시작점에 도달할 수 있는 지점입니다.

따라서 실제로 버팀줄이 처음 위치로 돌아온다는 사실을 알 수 있습니다.

그 다음으로, 버팀줄이 언제나 수열의 순서에 맞춰 존재하겠느냐?가 참임을 보이는 것은 어렵지 않습니다. 이 역시 수학적 귀납법을 활용하면 충분합니다. □

2.4 매듭 땋기

위에서 기록해둔 수열의 모임과 버팀줄을 가지고 실제로 매듭이 되는지 판별하는 알고리즘을 만들겠습니다.

1. 각 버팀줄에 서로다른 기호를 배당합니다.

2. 먼저 수열의 모임에서 수열을 하나 뽑습니다.

3. 수열의 첫 숫자부터 순서대로, 그 숫자의 앞에 반드시 위치해있을 두 버팀줄을 동일한 숫자의 반대 부호를 가진 뒤로 옮깁니다. (이는 위에서 보인 더 많은 보조정리들에서 증명한 사항입니다.)

   만약 옮기려고 하는 위치에 이미 기호가 배당되어 있는 경우, 이 기호가 서로 일치한다면, 계속하고

   일치하지 않는다면, 매듭이 아니라는 뜻입니다.

4. 이 과정을 숫자의 갯수에서 한번 더 반복합니다.

위 과정을 수행하면 우리는 주어진 가우스 코드가 매듭이냐 아니냐라는 문제 뿐만 아니라, 해당 매듭의 땋음 표현까지 구할 수 있습니다.

각 기호들은 땋음의 첫 시작 점을 표현하며 우리가 구해낸 수열의 모임은 각 교차점이 등장하는 순서와 같습니다. 이를 통해 땋음의 표현도 구할 수 있습니다.