사다리 타기의 진실

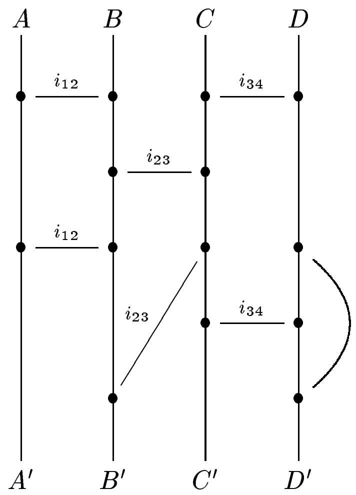
정 철주

Abstract

본 technical memo에서는 사다리 타기가 왜 one-one correspondence를 내주는지를, 사다리 타기는 결국 permuatation generation을 해주는 과정에 불과하다는 사실로서 증명한다.

1 서론

다음과 같은 사다리가 주어졌다고 하자.



도대체 사다리는 왜 A, B, C, D 각각에 대하여 하나의 결과만을 1대 1로 정확히 대응시켜주는가? 라는 의문을 가져본 사람들이 많을 것이다. 결론적으로 말하자면 '사다리'란 $ABCD$라는 ordered sequence에 대한 *permutation generator*에 불과하다는 사실이다. Permutation의 정의를 되새겨보면 1-1 대응이고 뭐고 별 대단한 것은 하나도 없다는 것을 알 수 있다.

2 용어 정의

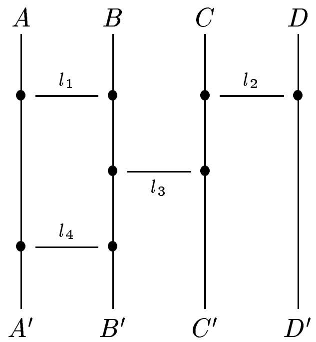
각 수직선 하나를 *line*이라고 부르기로 하자. 그리고 두 개의 서로 다른 line들을 잇는 직선을 *bridge*라 부르기로 한다. Bridge와 line이 만나는 점을 endpoint라 한다.

하나의 bridge b는 두 개의 endpoint로 표현될 수 있는데, 이 때, 왼쪽의 endpoint를 left point, 그리고 오른쪽의 endpoint를 right point라 정의하자. 이를

$$b = (b\text{의 left-point}, b\text{의 right-point})$$

로 나타내기로 한다.

3 원조 사다리



위의 사다리에는 4 개의 bridge l_1, l_2, l_3, 그리고 l_4가 있다. 이 bridge들 간에는 partial ordering $\leq$를 정의할 수 있는데, 이 $\leq$는 다음과 같이 정의된다: $l_i = (a, b)$이고 $l_j = (c, d)$라고 할 때, 다음의 조건을 만족시킬 때, $l_i \leq l_j$라고 정의한다:

(a) $b = c$이고 b, c 중 하나가 다른 하나의 immediate predecessor이다;

(b) $a = d$이고 a, d 중 하나가 다른 하나의 immediate predecessor이다.

이제, 이 partial ordering에 의거하여 다음과 같은 *bridge-ordering graph* $G = (V, E)$를 그릴 수 있다. G는 vertex set으로서 $V = \{l_1, l_2, \cdots, l_4\}$, 그리고 edge set으로서 $E = \{\langle l_i, l_j \rangle \mid l_i \leq l_j\}$이 된다.

위에 제시된 사다리에 대한 bridge-ordering graph G는 다음과 같다.

$$\begin{array}{ccc} l_1 & & l_2 \\ & \searrow \quad \swarrow & \\ & l_3 & \\ \swarrow & & \\ l_4 & & \end{array}$$

이제 G의 vertex set, $V(G)$에 대한 topological order를 구해보자.

$$\langle l_1, l_2, l_3, l_4 \rangle \text{ 또는 } \langle l_2, l_1, l_3, l_4 \rangle$$

가 $V(G)$에 대한 topological order가 된다. 이제, $ABCD$를 length 4의 ordered sequence라고 하자. *Inversion*이 ordered sequence 내에서 두 개의 원소의 위치를

바꾸는 연산이라고 정의할 때, *각 bridge는 하나의 inversion을 나타낸다는 사실을 알 수 있다.*

예를 들어, topological order $\langle l_1, l_2, l_3, l_4 \rangle$에 대한 ABCD의 permutation generation 과정은 다음과 같다.

$$\begin{array}{rcl} Initially & : & ABCD \\ l_1 \equiv i_{12} & : & BACD \\ l_2 \equiv i_{34} & : & BADC \\ l_3 \equiv i_{23} & : & BDAC \\ l_4 \equiv i_{12} & : & DBAC \end{array}$$

결국, 하나의 사다리는 inversion의 sequence를 정의하며 permutation이 일련의 inversion으로 정의될 수 있음을 상기할 때, 사다리는 하나의 permutation을 정의해 줌을 알 수 있다.

증명은 복잡하기만 하고 별다른 것이 없으므로 생략한다.

4 사다리는 진정한 permutation generator인가?

사다리는 모든 가능한 permutation을 생성해줄 수 있는 것인가? n 명에 대한 가능한 permutation의 수는 $n!$이다. 그렇다면 사다리를 이용하여 $n!$의 perumutation들을 모두 생성할 수 있는 것일까? 사다리의 bridge는 반드시 인접한 line 간에만 정의가 되기 때문에 이 문제는 결국 다음의 문제로 환원된다.

> Length n의 sequence의 임의의 두 원소의 위치를 바꾸고자 한다. 일련의 consecutive 원소들 간의 swapping 만으로 이 작업이 가능한가?

답부터 말하자면, '가능하다'라는 것이다. 예를 들어, $S = s_1 s_2 \cdots, s_n$이라는 sequence가 주어졌다고 하자. s_i와 s_j를 $(1 \leq i < j \leq n)$ 서로 맞바꾸는 permutation은 subsequence $S' = S(i:j)$에 대하여 다음의 알고리즘을 사용하면 된다.

$|S'| = j - i + 1 \equiv 0 (\mathrm{mod}\ 2)$, 즉, S'의 길이가 짝수인 경우에 대해서만 설명하겠다. $|S'|$가 홀수인 경우도 비슷하게 해결할 수 있다.

LONGDISTANCESWAP(S, i, j)

$n' \leftarrow j - i + 1$
// 견우, 오작교에 도착하다
for $t = i$ **to** $\lfloor n'/2 \rfloor$
do swap s_i and s_{i+1}
// 직녀, 오작교에 도착하다
for $t = n$ **downto** $\lceil n'/2 \rceil$
do swap s_i and s_{i-1}
swap $s_{\lfloor n'/2 \rfloor}$ and $s_{\lceil n'/2 \rceil}$
// 직녀는 견우의 집으로
for $t = i$ **to** $\lfloor n'/2 \rfloor$
do swap s_i and s_{i+1}

// 견우는 직녀의 집으로
for $t = n$ **downto** $\lceil n'/2 \rceil$
do swap s_i and s_{i-1}

결국, 우리는 constructive하게 사다리는 진정한 permutation generator임을 증명한 셈이 된다.