

# Lectures on Tensor Categories and Modular Functor 学习笔记

X. D. H.

2025 年 8 月 2 日

## 摘要

这个笔记我希望 follow 2000 年的书籍「Lectures on Tensor Categories and Modular Functor」学习三个领域之间的关系。CFT-TQFT-Tensor Category。理清楚他们之间到底是什么联系捏。我需要再尝试一下下

# 目录

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>QFT in Functors</b>	<b>4</b>
<b>3</b>	<b>TQFT</b>	<b>5</b>
<b>4</b>	<b>Scratch Book</b>	<b>6</b>
4.1	Chiral Vertex Operator . . . . .	6
4.1.1	Definition of Chiral Vertex Operator . . . . .	6
4.1.2	Chiral Vertex Operator and Conformal Block . . . . .	7
4.2	Modular Tensor Categories description of RCFT . . . . .	8

# Chapter 1

## Introduction

我们的目标是理解三个不同的对象之间的联系：

- Tensor Category
- 3D TQFT
- 2D Modular Functors

我们先简要的介绍一下三个主要的对象分别定义是什么并且有什么关系。

1. **Tensor Category** 是一种 abelian 的范畴；并且包含一些性质「满足结合律的张量积结果，存在 unit object, rigid (每个 object 存在 dual object)」

我们注意：物理上我们只考虑 semi-simple 的这样的范畴。（如，rep of compact group 范畴）；但是不像 group-rep 范畴这么简单

### Tip: 从群表示范畴的推广

我们一般群表示范畴会存在一个交换映射： $\sigma_{WV} : W \otimes V \rightarrow V \otimes W$ ，其中  $W, V$  是表示空间。这个映射是一个线性映射把两个表示空间张量积之后的向量，映射到反顺序张量积之后的向量。

对于一般群表示来说这个映射根据定义就是  $\sigma^2 = 1$  成立的，但是对于量子群来说，为了群元素作用其上是合法的，我们并不能够这么单纯的定义  $\sigma$  映射。所以并不一定满足  $\sigma^2 = 1$  的关系。

此外我们一般考虑一种特殊的 tensor category 也就是 modular tensor category。满足一些额外的性质和结构

2. **TQFT**，是一个数学结构。一个单纯的定义是：「一种赋予流形  $M$  一个线性空间  $\tau(M)$ ；同时赋予一个 cobordism 一个线性映射的法则」当然这个定义还可以推广，我们不一定考虑单纯的流形。我们可以在二维流形上赋予一些 marked point 的结构；给三维流形赋予一些 tangle 在里面。我们会考虑这些 extended TQFT。
3. **2D Modular Functor**：反正我是没看懂，建议之后好好学习呃呃呃。有一点感觉但是不是很懂。

对于 modular functor 存在两种定义「topological 以及 complex-analytic」对于 complex analytic 的 modular functor 的结构在所有的 RCFT 之中都存在!!

### Important: 本书重要结论

Modular Tensor Categorg & 3D TQFT & 2D Modular Functor 其实结构是等价的!!

#### quantum Yang-Baxter equation

下面给出一个简单的例子说明一下这三个东西是什么。我们分别用三个体系的语言描述同样的 quantum Yang-Baxter Equation 这个东西。

- Modular Tensor Category 语境

我们可以定义一个 semi-simple Abelian Category  $\mathcal{C}$  并且认为这个 category 有着下面的一些 simple objects  $V_1 \dots V_n \in \mathcal{C}$ 。

在此基础上我们可以定义这个 category 的一个特殊的 morphism, 由于是 tensor category, 我们允许 object 的张量积是 object 并且 object 之间存在 morphism 所以我们定义 commutativity isomorphisms 为 (之后我们也会用  $\sigma_i$  进行标记):

$$\sigma_{V_i V_{i+1}} : V_1 \otimes \dots \otimes V_i \otimes V_{i+1} \otimes \dots \otimes V_n \rightarrow V_1 \otimes \dots \otimes V_{i+1} \otimes V_i \otimes \dots \otimes V_n. \quad (1.1)$$

对于 Tensor Category 的公理体系需要保证这些 isomorphism 存在一些约束要求:

$$\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1} \quad (1.2)$$

这个约束条件就是 【quantum Yang-Baxter equation】。

- 3D TQFT 语境

我们考虑一个 2-sphere 的结构, 也就是  $S^2 = \mathbb{R}^2 \cup \infty$  并且这个球上面有 n 个 marked point 也就是  $p_1 = (1, 0) \dots p_n = (n, 0)$  就是这些在实数轴上面的点。并且这些的 point 上面我们分别赋予一个抽象的 object  $V_1 \dots V_n$ 。对于一个这样的特殊结构的流形上面我们可以通过 3D TQFT 的方法构建一个向量空间。

此外我们考虑一个这样的三维流形  $M = S^2 \times [0, 1]$  对于这样的流形我们可以考虑如果其内部有这样的狗性的 tangle, 那么这个 TQFT 的规则应该给出了一个线性映射:

$$\sigma^{\text{TQFT}} : \tau(S^2; V_1, \dots, V_n) \rightarrow \tau(S^2; V_1, \dots, V_{i+1}, V_i, \dots, V_n) \quad (1.3)$$

构型如下图所示:

# Chapter 2

## QFT in Functors

# Chapter 3

## TQFT

# Chapter 4

## Scratch Book

这里会放一些写的很混沌，但懒得扔掉的东西呜呜呜呜!!! 这里会放一些写的很混沌，但懒得扔掉的东西呜呜呜呜!!! 这里会放一些写的很混沌，但懒得扔掉的东西呜呜呜呜!!! 这里会放一些写的很混沌，但懒得扔掉的东西呜呜呜呜!!!

### 4.1 Chiral Vertex Operator

我们研究一个量子场论最重要的一个东西就是量子场，也就是我们的关联函数。对于二维的共形场论求解关联函数可以通过一个“积木”来实现。也就是 conformal block。可以像是一点点搭积木一样拼凑成为一个二维的共形场论的关联函数。

对于共形场论来说，我们一般求解的“场”。其实就是 vertex operator。因为，这个算符或者说场对应了 Virasoro 代数的表示的量子态。我们知道一个量子态其实是希尔伯特空间的一个向量。对于共形场论来说希尔伯特空间是 Vir 代数的表示的直和。

在认识到上面基本的知识之后，我们接下来介绍我们的目的。我们要研究 RCFT 之中包含的一个特殊的数学结构。这个数学结构使得我们的对偶是可能的。其中很重要的一个两就是我们的 Chiral Vertex Operator。

#### 4.1.1 Definition of Chiral Vertex Operator

接下来我们一点点定义我们的 Chiral Vertex Operator，这里的定义我们只考虑 minimal model 的情况，但是 RCFT 并不仅仅只有 minimal model。对于 affine Lie Algebra 的定义我们后面会进行讨论：

**Definition 1.** *Chiral Vertex Operator*

- CVO 是一个从 Vir 代数的表示到另一个 Vir 代数的表示的映射：

$$\Phi_{i,k}^{j,\beta}(z) : H_i \rightarrow H_k \quad (4.1)$$

这个算符是由另一个表示空间  $j$  之中的一个量子态  $\beta$  标记的。从表示空间  $i$  到表示空间  $k$  的映射

- 对于 *primary state* 标记的 *CVO* 的矩阵元素，当对于 *primary state* 的元素。我们有定义：

$$\langle i | \Phi_{i,k}^j(z) | k \rangle = \|\Phi_{i,k}^j\| z^{-(\Delta_j + \Delta_k - \Delta_i)} \quad (4.2)$$

- 对于 *primary state* 标记的 *CVO* 的矩阵元素，当对于 *descendents* 的元素。我们可以求解：

$$[L_n, \Phi_{ik}^{j,\beta}(z)] = \left( z^{n+1} \frac{d}{dz} + (n+1)z^n \Delta(\beta) \right) \Phi_{ik}^{j,\beta}(z). \quad (4.3)$$

方程得到。

- 对于 *descendent* 标记的 *CVO* 我们可以直接把  $L_{-\mathcal{I}}$  作用在 *CVO* 上面（因为，*CVO* 可以认为是 *Vir* 代数表示对应的量子场）得到：

$$\Phi_{ik}^{j,\beta}(z) = L_{-\mathcal{I}}^{(z)} \Phi_{ik}^j(z) \quad (4.4)$$

其中  $\Phi_{ik}^j(z)$  指 *primary field* 标记的 *CVO*。此外对于 *V* 代数作用在场上面其实就是进行一个留数积分：

$$\Phi_{ik}^{j,\beta}(z) \equiv \oint d\xi_1 (\xi_1 - z)^{n_1+1} T(\xi_1) \dots \oint d\xi_\ell (\xi_\ell - z)^{n_\ell+1} T(\xi_\ell) \Phi_{ik}^{j,j}(z). \quad (4.5)$$

下面是一个简单的计算的例子，可以熟悉相关 *V* 代数的计算。我们计算一个  $\Delta_\phi \neq 0$  的表示的 *CVO* 在真空态下面的元素：

$$\langle 0 | [L_{-1}, \Phi_{0,0}^\phi] | 0 \rangle = \partial_z \langle 0 | \Phi_{0,0}^\phi | 0 \rangle = \left\| \Phi_{0,0}^\phi \right\| (-\Delta_\phi) z^{-\Delta_\phi - 1} = 0 \quad (4.6)$$

最后等于 0 是因为，我们已知真空态是  $\Delta = 0$  的表示的 *primary state*。这个其实它自己也是一个 degenerate state。满足关系  $L_{-1} | 0 \rangle = 0$ 。因此可以求出来  $\left\| \Phi_{0,0}^\phi \right\| = 0$ 。

### Remark:

我们为什么要定义 *CVO*? 首先我们明确 *CVO* 并没有实际的物理意义。对于一个算符，我们不可能把 holomorphic 和 anti-holomorphic 的部分分开。但是，我们一个很简单的观察是，对于 CFT 的三点函数来说，计算结果 holomorphic 和 anti-holomorphic 的部分是可以分开的。

因此，我们定义 *CVO*，本质上就是定义一种等价于 conformal block 的算符理论。更准确的说是相当于把三点函数的 conformal block 进行一个算符意义上面的推广。

### 4.1.2 Chiral Vertex Operator and Conformal Block

下面我们讨论，我们这样定义这个算符怎么帮助我们研究 conformal block 和他们之间的关系。

## 4.2 Modular Tensor Categories description of RCFT

本章我们给出一个很强的结论!

**Theorem 1.** *RCFT 是群理论的推广*

我们认为 *RCFT* 对应着 *Modular Tensor Category*; 群理论对应着 *Tannaka Category*。而 *Modular Tensor Category* 是 *T Category* 的推广。因此我们认为, *RCFT* 是群理论的推广!

为了说明这个定理, 我们首先考虑群理论的结构的另一种表达, 称为 Tannaka-Klein Theory 为此我们考虑的一个对象是一个群的所有有限维度的表示:

$$\text{Rep}(\mathcal{G}) = \{V \mid V \text{ is finite dimensional representation of } \mathcal{G}\} \quad (4.7)$$

我还想写一点测试的话