# Lie Theory 0x02

DaeYong Kim

Dept. of Artificial Intelligence, Ajou University

# Contents

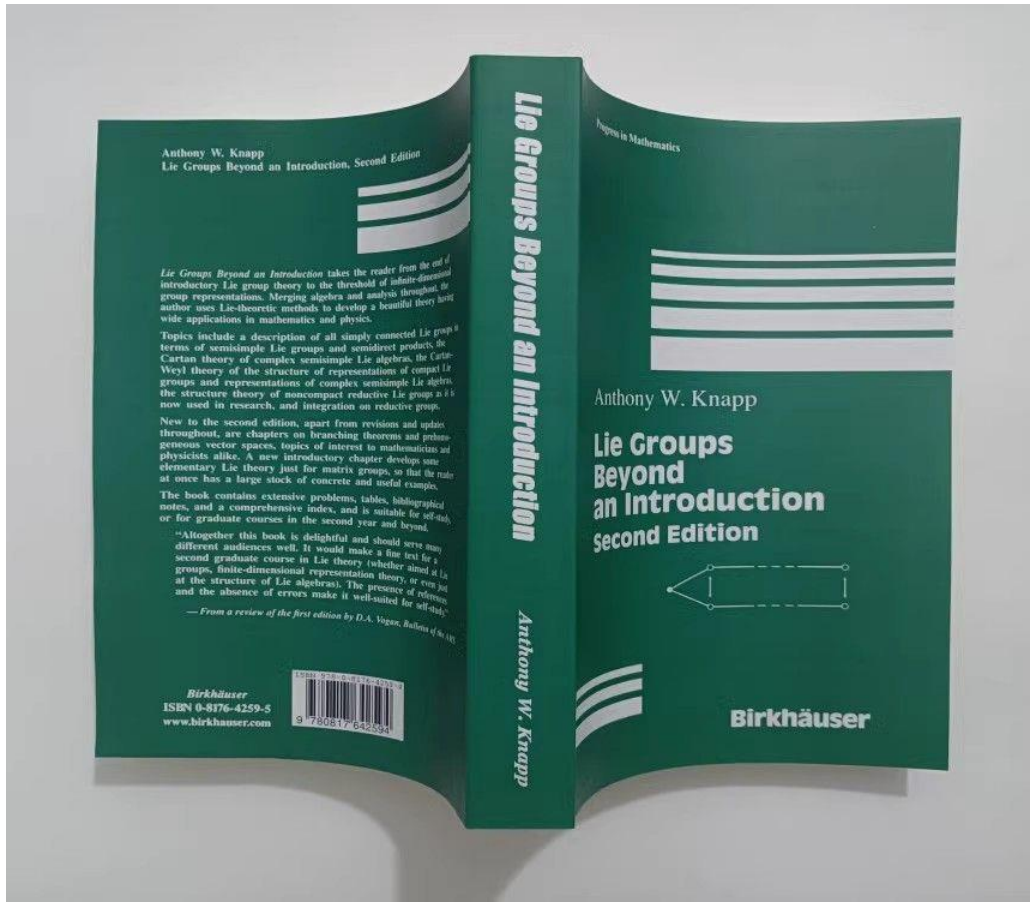
## Algebraic System

- What is the Algebraic System

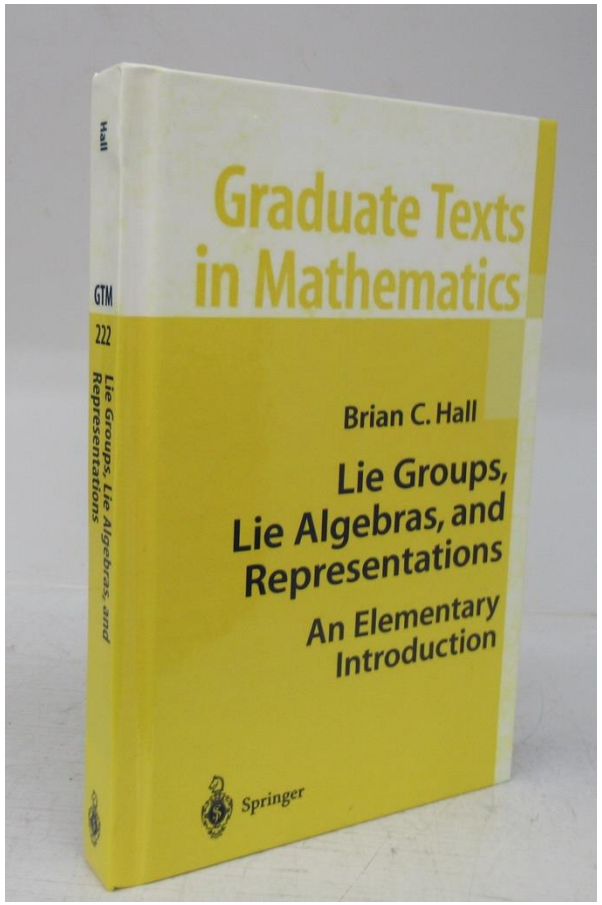
## Lie Theory

- What is the Lie Theory
- Group theory
- Properties
- Set && Space && Group
- group action
- Features of Lie Group
- Lie Group and Lie Algebra
- Mapping: Exponential and Logarithmic Operation

## References



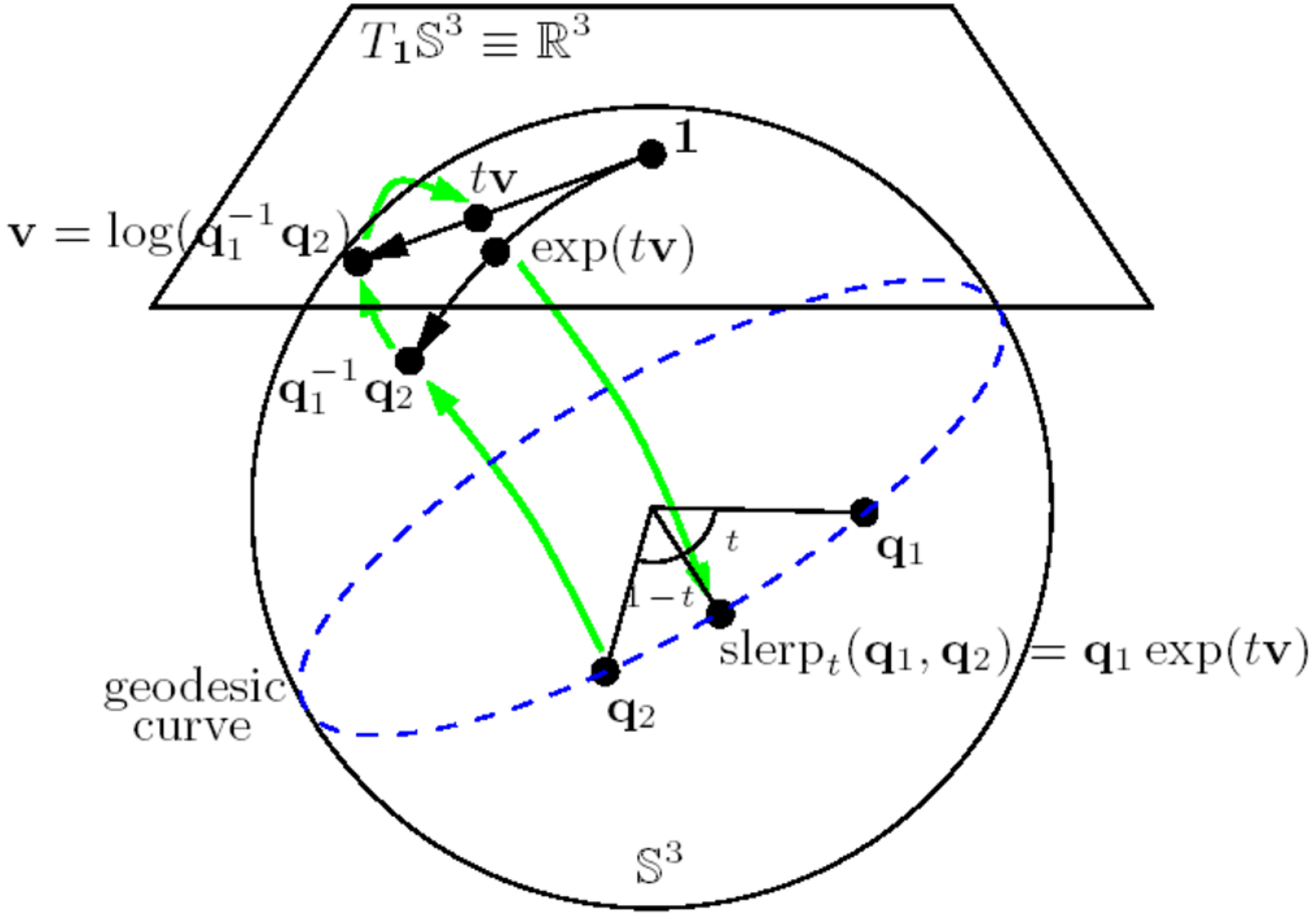

**Lie Groups: Beyond an Introduction 2nd Edition by Antony W. Knapp**




**Lie Groups, Lie Algebras, and Representations: An Elementary Introduction (Graduate Texts in Mathematics, 222) 2nd ed. 2015 Edition**

# Preview: Spherical Linear Interpolation

## Goal



# Algebraic System

# Is “the Theory of everything” merely the ultimate ensemble theory?

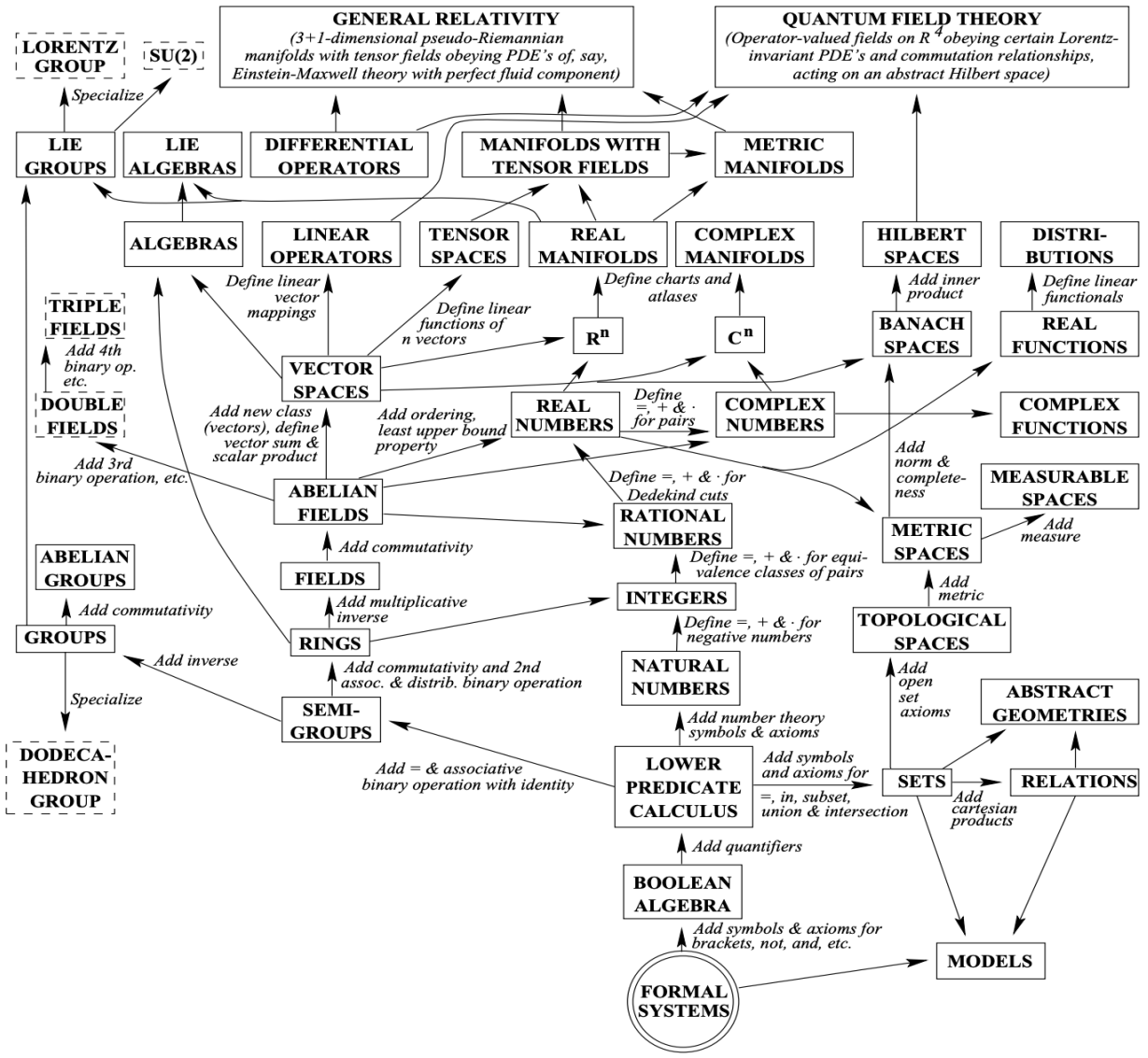
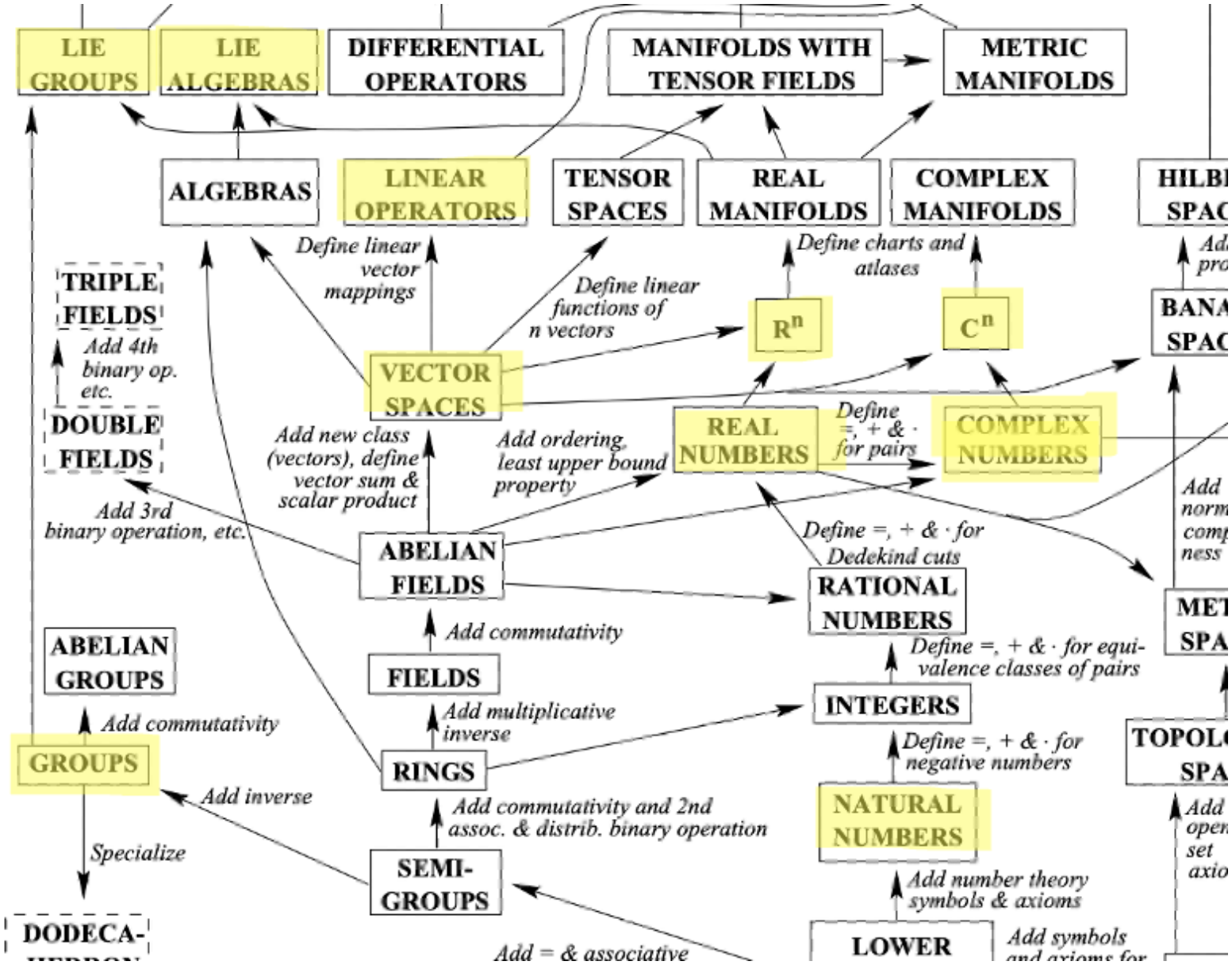



FIG. 1. Relationships between various basic mathematical structures. The arrows generally indicate addition of new symbols and/or axioms. Arrows that meet indicate the combination of structures — for instance, an algebra is a vector space that is also a ring, and a Lie group is a group that is also a manifold.

# Is “the Theory of everything” merely the ultimate ensemble theory?



# What is the Algebraic System

**대수적 체계(Algebraic system)란 무엇인가 그리고 어떻게 정의되는가**

대수적 체계: 집합, 하나 이상의 연산 + 연산들이 만족해야 하는 공리로 구성된 수학적 구조

Set(집합)
- 대수적 체계에서 다루고자 하는 대상들의 모임
- e.g. 자연수, 정수, 실수, 복소수, …

Operation(연산)
- 집합의 원소들 사이에서 정의된 함수
- 원소들을 결합하여 새로운 원소로 매핑
- e.g. Binary operation, Unary Operation, …

Axiom or Properties(공리 또는 성질)
- 연산이 만족해야 하는 규칙이나 법칙
- 체계의 구조를 결정

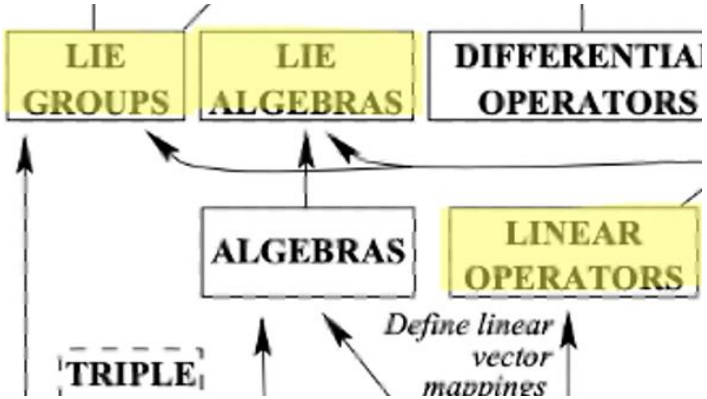
# Lie Theory

# What is the Lie Theory

## Lie Theory

- 변환군 관련 이론
- 사용 분야: 양자역학, 수학, 시스템 제어, 컴퓨터 그래픽스 등

Marius Sophus Lie(1842 – 1899): Norwegian mathematician.




# What is the Lie Theory

## Group Theory: 군론

Group(군): **특정 조건들**을 만족하는 **대수적 구조**

Set(집합): A = $\{a_1, a_2, a_3, a_4, \dots, a_n\}$
Operation(연산): *

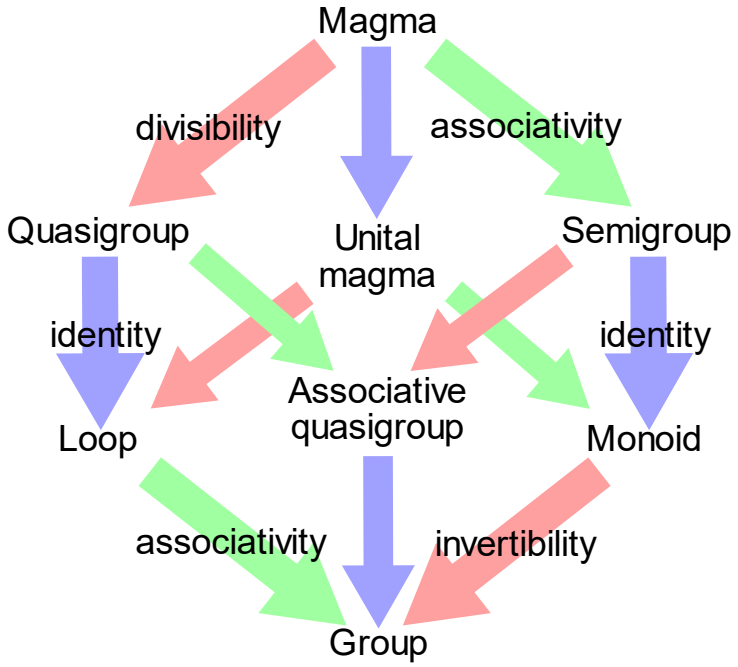
G = (A, *)

닫혀있다: $a_i$ (operation) $a_j$ => $a_k$ ∈ A

### e.g.) 루빅스 큐브

G = (A, *)
A = 가능한 모든 회전의 집합
* = 움직임의 합성

닫혀있다: $a_i$ * $a_j$ => $a_k$ ∈ A




<대수적 구조>

루빅스 큐브를 돌리는 방법들을 모은 집합은 군을 이룬다.

# What is the Lie Theory

## Examples: Set && Operation

| Group | Set elements | Operation | e.g. |
|---|---|---|---|
| ℕ(자연수) | 1, 2, 3, … | + | 1 + 1 = 2 |
| ℤ(정수) | -2, -1, 0, 1, 2, 3, … | -, +, x | (-6) + 2 = -4 |
| ℝ(실수) | 1/3, 0.5, 0, 2, … | -, +, x, / | 1 / 2 = 0.5 |

## 닫혀 있지 않는 경우

ℕ: 1 – 2 = -1 → 자연수는 (-)연산에 대해서 닫혀 있지 않음

ℤ: 1 / 2 = 0.5 → 정수는 나눗셈에 대해서 닫혀 있지 않음

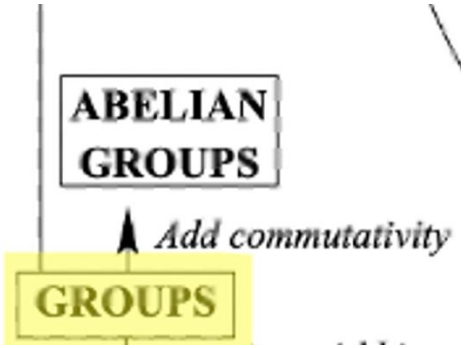
# What is the Lie Theory

## Main Axioms || Properties

- Closure(폐쇄성): 연산 결과가 항상 집합 내에 존재해야 함
  $\forall a, b \in G, a * b \in G$
- Associativity(결합법칙): (a+b)+c = a+(b+c)
- Identity Element(항등원): 집합 G에는 항등원 e가 존재
  $\exists e \in G, \forall a \in G, e * a = a * e = a$
- Inverse Element(역원): 각 집합 G의 원소에 대해서 역원이 존재
  $\forall a \in G, \exists a^{-1} \in G, a * a^{-1} = a^{-1} * a = \mathrm{e}$

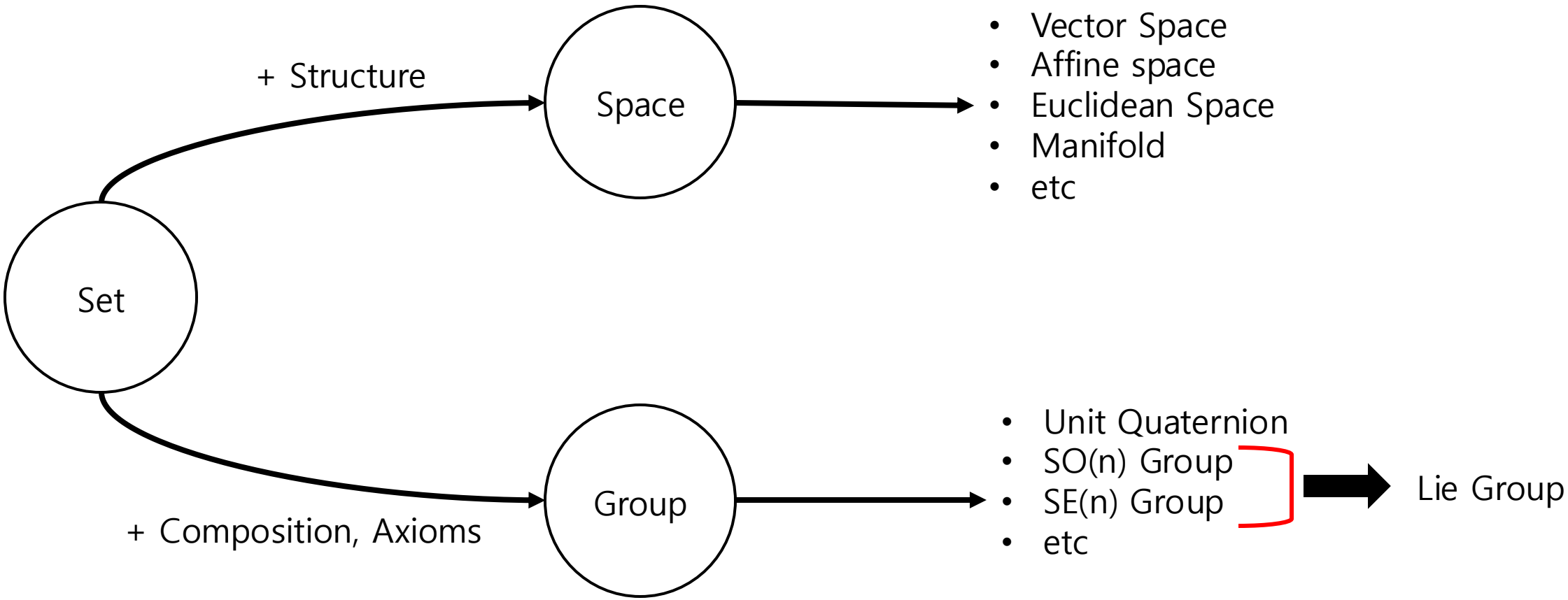
Axioms of Group

+ Commutativity(교환법칙): a*b = b*a



# Set && Space && Group

## Relationship



# Group Action

## Group Action

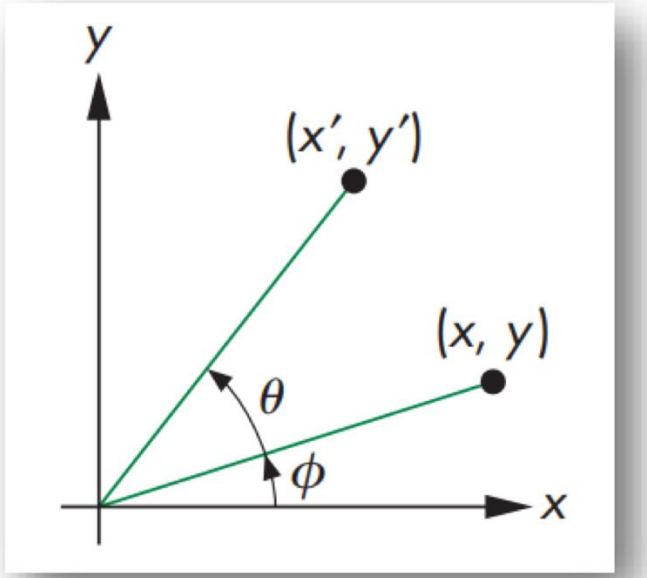
- 또 다른 집합 또는 군을 변환(=act) 시킬 수 있다.
- 군이 특정 집합을 변환하는 연산자 역할을 할 수 있다는 것을 의미
- Lie Group은 3차원 공간 상에서 물체의 이동+회전을 표현하기 적합한 도구

## Lie Group

- SO(n) : Rotation Matrix
- SE(n): Transformation Matrix

# Example

## Group Action of SO(2): Rotation Matrix




$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\boldsymbol{x}' = \mathbf{R} \cdot \boldsymbol{x}$$

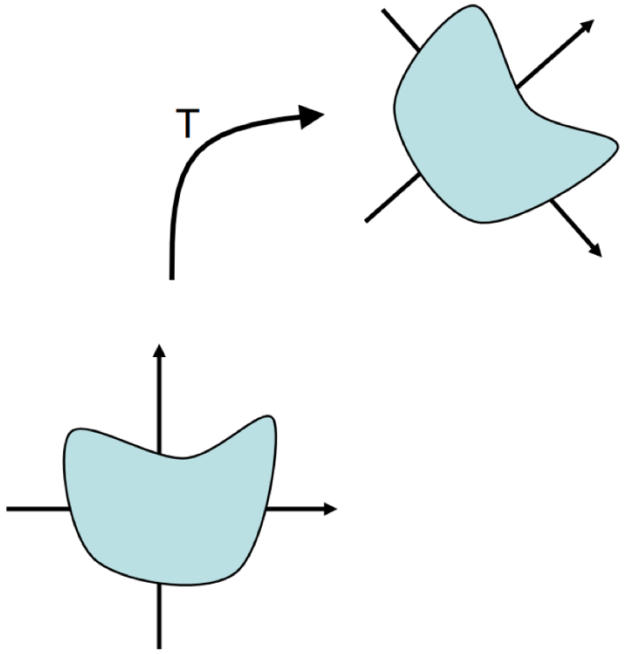
(Vector Space) = (Lie Group)**(Composition)**(Vector Space)

회전행렬: $\mathbf{R} \in SO(2)$
2차원 벡터: $\mathbf{x} \in \mathbb{R}^2$
이항연산: $\cdot$

$\mathbf{R}$은 vector space의 한 점을 회전(=act)

# Example

## Group Action of SE(3): Transformation Matrix




$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix}$$

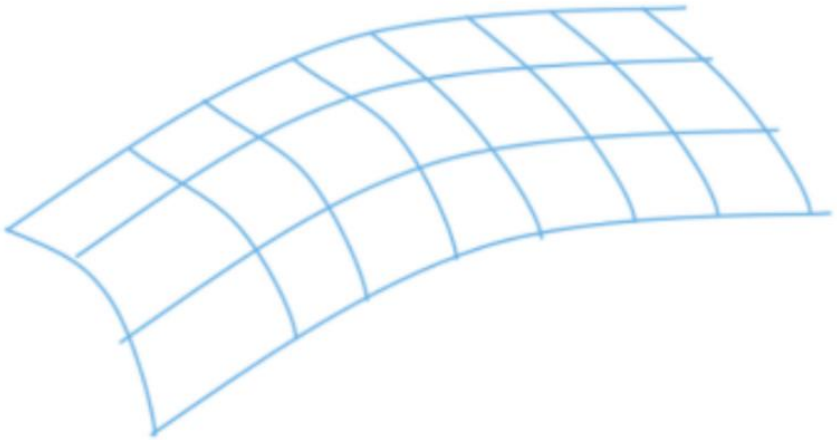
$$\boldsymbol{x' = T \cdot x}$$

변환행렬: $\mathrm{T} \in SE(3)$
2차원 벡터: $\mathbf{x} \in \mathbb{R}^3$
이항연산: $\cdot$

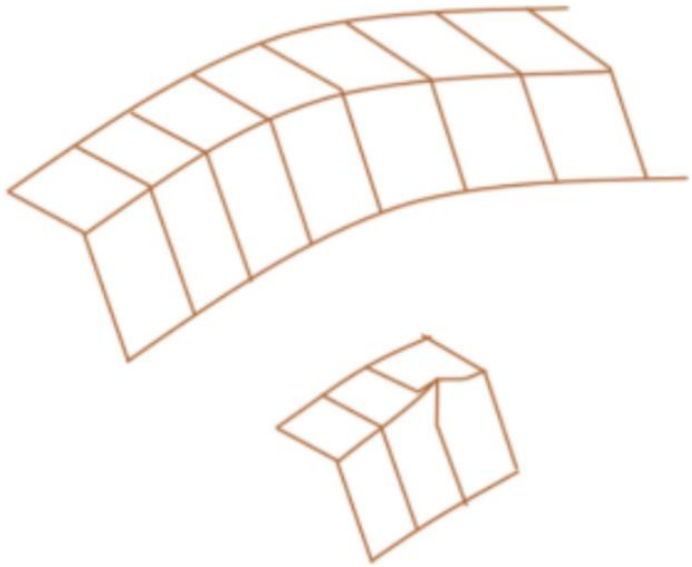
$\boldsymbol{T}$는 vector space의 한 점을 Rotation && Translation (=act)

# Features of Lie Group

## Smooth Manifold

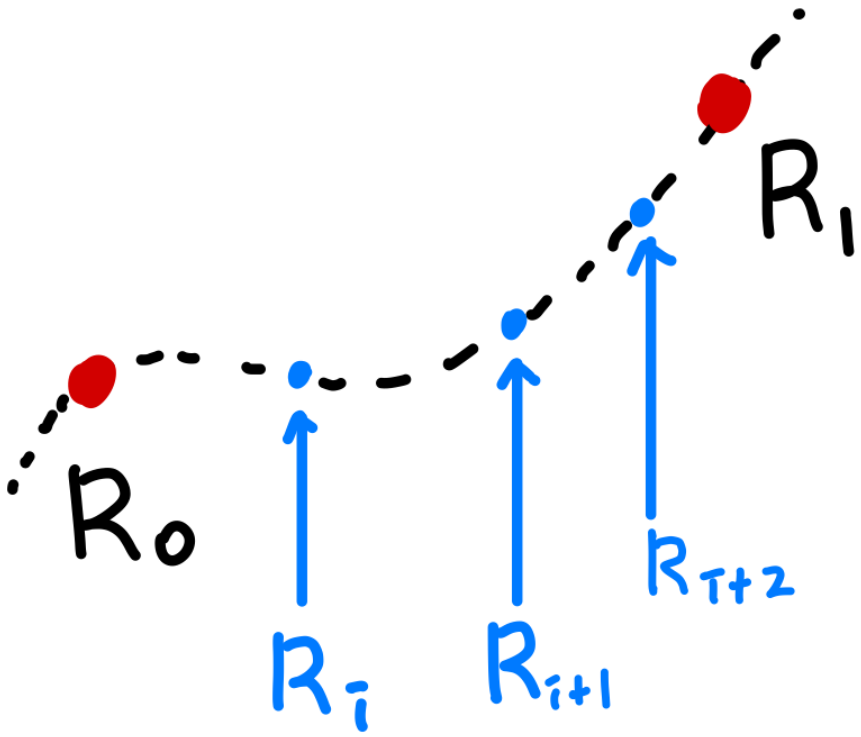


Smooth Manifold

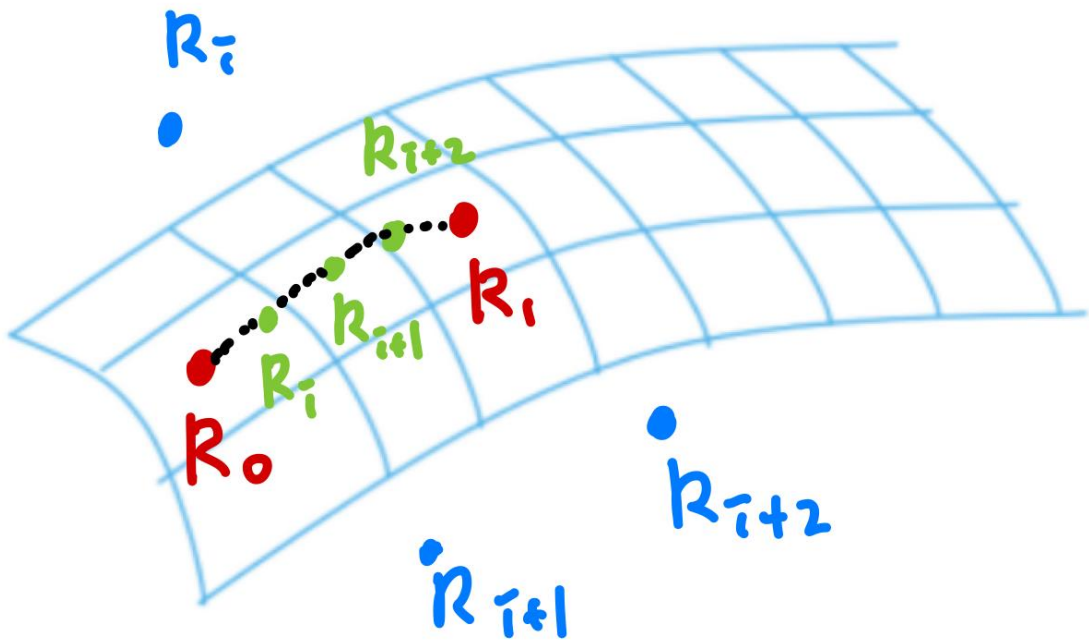


Non-Smooth Manifold (Edge, Spike)

# Features of Lie Group



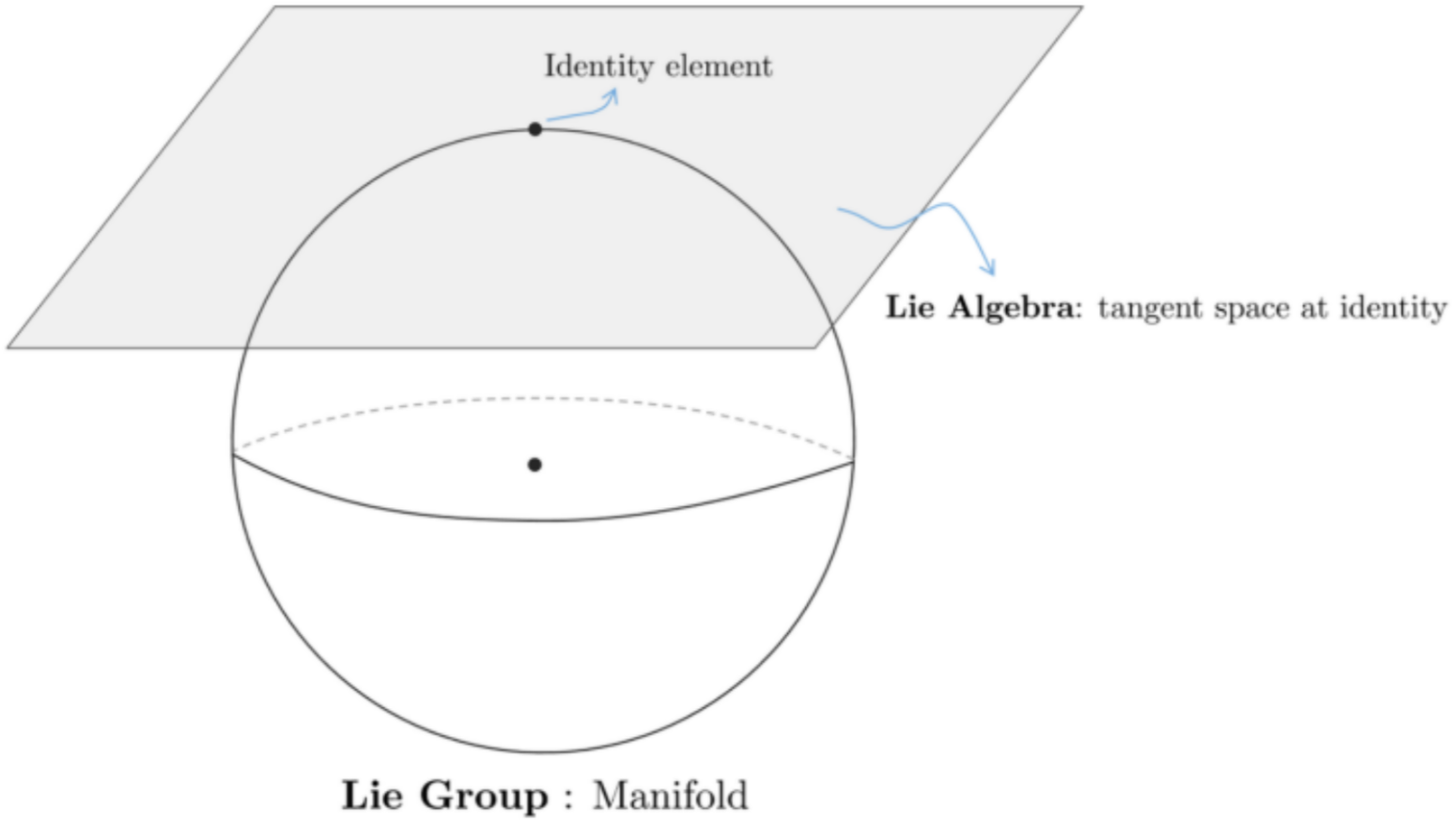

<What We Want>



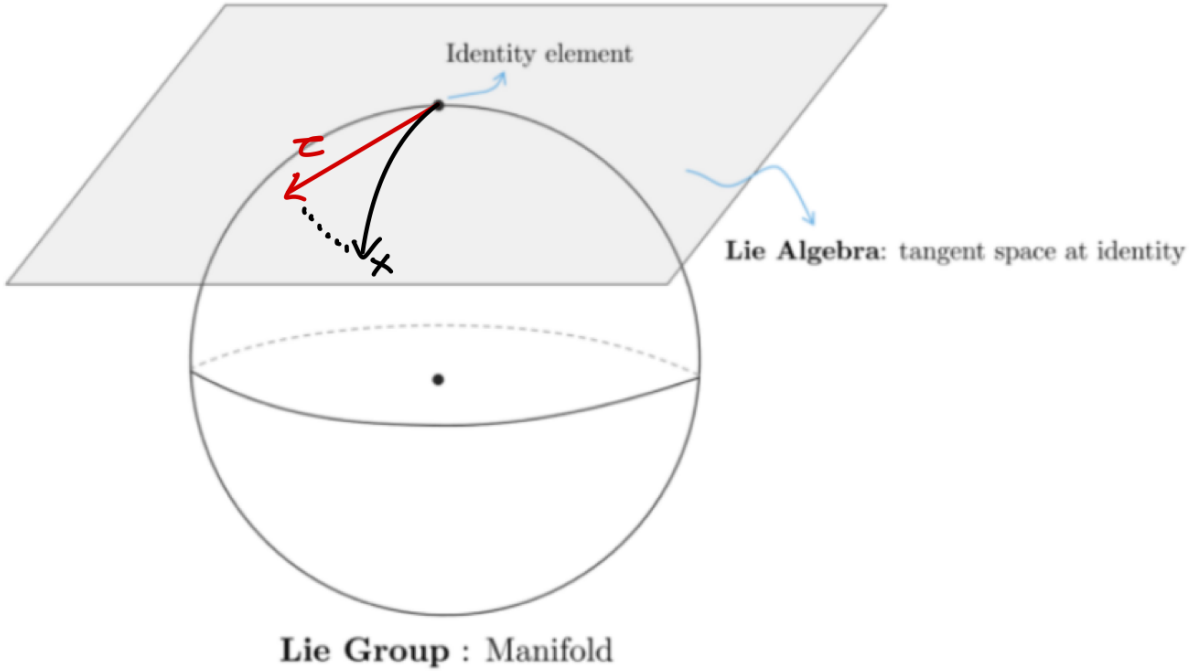

<Manifold>

# Lie Group and Lie Algebra

## Topological Structure: Lie Group, Lie Algebra



# Mapping: Exponential and Logarithmic Operation

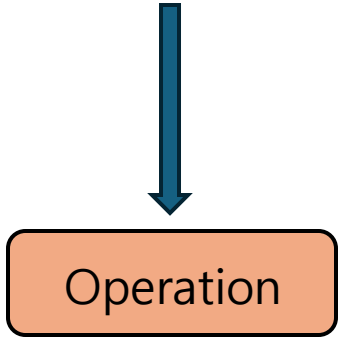



Lie Group : 까다로운 제약 조건
Lie Algebra: 비교적 자유로운 제약 조건

1대1 mapping: Lie Group ←→ Lie Algebra

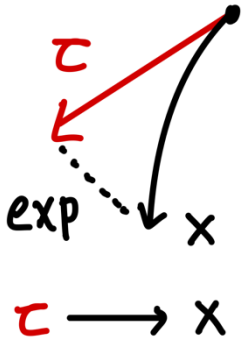
**연산 과정**

Lie Group → Lie Algebra → Lie Group

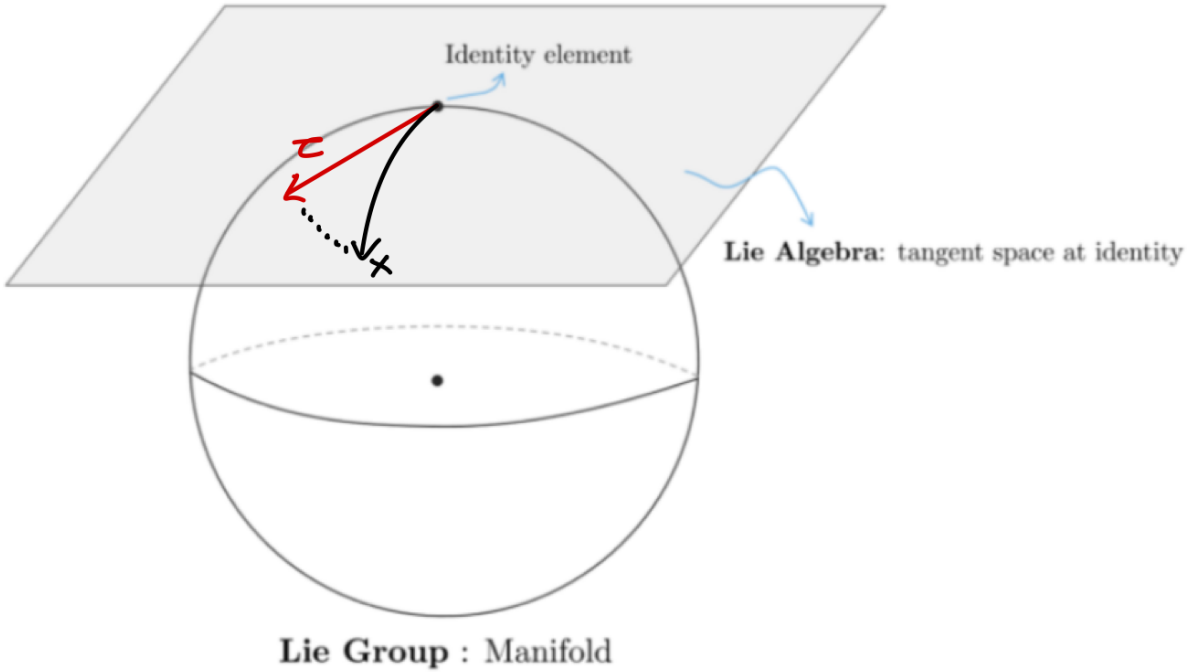


# Mapping: Exponential and Logarithmic Operation

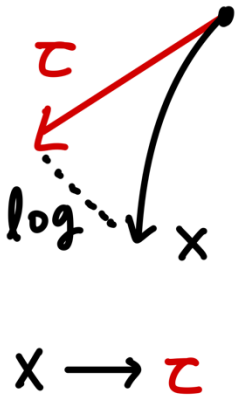
## Exponential Mapping




Lie Algebra → Lie Group




## Logarithmic Mapping




Lie Group → Lie Algebra

# References

https://slideplayer.com/slide/16959877/

https://www.sciencedirect.com/science/article/pii/S0003491698958559

https://en.wikipedia.org/wiki/Group_theory

https://drive.google.com/viewerng/viewer?url=https://github.com/gyubeomim/gb-supp-mat/blob/main/blog/Notes+on+Lie+Theory.pdf?raw%3DT

# Q&A

# Bonus Lecture

# 초보자를 위한 생성형 AI 무료강좌 - 마이크로소프트


**21 Lessons teaching everything you need to know to start building Generative AI applications**

license MIT | contributors 93 | issues 12 open | pull requests 9 open | PRs welcome

https://github.com/microsoft/generative-ai-for-beginners?tab=readme-ov-file

# 초보자를 위한 생성형 AI 무료강좌 - 마이크로소프트

## Generative AI for Beginners (Version 3) - A Course

Learn the fundamentals of building Generative AI applications with our 21-lesson comprehensive course by Microsoft Cloud Advocates.

### 🌱 Getting Started

This course has 21 lessons. Each lesson covers its own topic so start wherever you like!

Lessons are labeled either "Learn" lessons explaining a Generative AI concept or "Build" lessons that explain a concept and code examples in both **Python** and **TypeScript** when possible.

Each lesson also includes a "Keep Learning" section with additional learning tools.

### What You Need

#### To run this code of this course, you can use either:

- Azure OpenAI Service - **Lessons:** "aoai-assignment"
- GitHub Marketplace Model Catalog - **Lessons:** "githubmodels"
- OpenAI API - **Lessons:** "oai-assignment"
- Basic knowledge of Python or TypeScript is helpful - *For absolute beginners check out these Python and TypeScript courses.
- A GitHub account to fork this entire repo to your own GitHub account

We have created a **Course Setup** lesson to help you with setting up your development environment.

Don't forget to star (🌟) this repo to find it easier later.

# 초보자를 위한 생성형 AI 무료강좌 - 마이크로소프트

| # | Lesson Link | Description | Video | Extra Learning |
|---|---|---|---|---|
| 00 | Course Setup | **Learn:** How to Setup Your Development Environment | Coming Soon | Learn More |
| 01 | Introduction to Generative AI and LLMs | **Learn:** Understanding what Generative AI is and how Large Language Models (LLMs) work. | Video | Learn More |
| 02 | Exploring and comparing different LLMs | **Learn:** How to select the right model for your use case | Video | Learn More |
| 03 | Using Generative AI Responsibly | **Learn:** How to build Generative AI Applications responsibly | Video | Learn More |
| 04 | Understanding Prompt Engineering Fundamentals | **Learn:** Hands-on Prompt Engineering Best Practices | Video | Learn More |
| 05 | Creating Advanced Prompts | **Learn:** How to apply prompt engineering techniques that improve the outcome of your prompts. | Video | Learn More |
| 06 | Building Text Generation Applications | **Build:** A text generation app using Azure OpenAI / OpenAI API | Video | Learn More |
| 07 | Building Chat Applications | **Build:** Techniques for efficiently building and integrating chat applications. | Video | Learn More |
| 08 | Building Search Apps Vector Databases | **Build:** A search application that uses Embeddings to search for data. | Video | Learn More |
| 09 | Building Image Generation Applications | **Build:** A image generation application | Video | Learn More |
| 10 | Building Low Code AI Applications | **Build:** A Generative AI application using Low Code tools | Video | Learn More |
| 11 | Integrating External Applications with Function Calling | **Build:** What is function calling and its use cases for applications | Video | Learn More |
| 12 | Designing UX for AI Applications | **Learn:** How to apply UX design principles when developing Generative AI Applications | Video | Learn More |
| 13 | Securing Your Generative AI Applications | **Learn:** The threats and risks to AI systems and methods to secure these systems. | Video | Learn More |
| 14 | The Generative AI Application Lifecycle | **Learn:** The tools and metrics to manage the LLM Lifecycle and LLMOps | Video | Learn More |
| 15 | Retrieval Augmented Generation (RAG) and Vector Databases | **Build:** An application using a RAG Framework to retrieve embeddings from a Vector Databases | Video | Learn More |
| 16 | Open Source Models and Hugging Face | **Build:** An application using open source models available on Hugging Face | Video | Learn More |
| 17 | AI Agents | **Build:** An application using an AI Agent Framework | Video | Learn More |
| 18 | Fine-Tuning LLMs | **Learn:** The what, why and how of fine-tuning LLMs | Video | Learn More |
| 19 | Building with SLMs | **Learn:** The benefits of building with Small Language Models | Video Coming Soon | Learn More |
| 20 | Building with Mistral Models | **Learn:** The features and differences of the Mistral Family Models | Video Coming Soon | Learn More |
| 21 | Building with Meta Models | **Learn:** The features and differences of the Meta Family Models | Video Coming Soon | Learn More |

# 초보자를 위한 생성형 AI 무료강좌 - 마이크로소프트