Discrete Mathematics

Siwon Yun

March 14, 2025

Contents

Theme 1
강의에 들어가며

컴퓨터 공학 전공하지 마라. 생물학 배우고파

— Jensen Huang, NVIDIA CEO

1.1 이산 수학이란

DEFINITION 1.1 (이산 수학)**.** 이산 수학discrete mathematics은 연속적인 대상이 아닌 이산적인 대상을 다루는 수학의 한 분야이다.

여기서 이산이란 분리 혹은 불연속으로 서로 구별될 수 있는 것을 의미하는데, 예컨대 graph는 모든 node와 edge는 서로 구별 가능하며 집합으로 표현 가능하다. 즉, 이산 수학은 **finite**하고 **countable** 한 것을 다루게 된다.

1.2 이산 수학을 왜 배우는가?

강의 자료에는 2가지 이유를 들고 있다. 1) 수학적 논리를 익혀 현실 세계에 대한 문제 해결력 향상. 2) 컴퓨터 분야의 자료 구조, 알고리즘 등 개념 확립.

이거 ChatGPT가 적은 것일까요, 아닐까요?

— 교수님

개인적으로 결과에는 원인과 동기가 굉장히 중요하다고 생각한다. 내가 이산 수학을 배우는 것은(결과) 공부의 필요성을 느끼고 중요함을 인지한(원인, 동기) 이후이길 바란다. 내가 요컨대 이산 수학을 배우는 이유는 다음과 같다:

NOTE 1.1. 이산 수학을 배우는 이유:

1. 컴퓨터는 이산적인 문제 해결에 특화되어 있음.
2. 때문에 컴퓨터를 활용한 문제 해결을 위해 이산 수학을 공부해야함.
3. 추가적으로 이산 수학 학습은 문제 해결 능력 향상에 필요함.

컴퓨터를 좋아하는 컴퓨터 공학부 학생에게 이정도면 이 과목을 배우는 충분한 이유가 될 듯 하다.

1.3 이산 수학 예시

강의에서 2가지 이산 수학 예시가 소개 되었다. 1) 쾨니스베르트의 다리 문제. 2) 하노이의 탑 문제.

1. 마치 한붓 그리기와 같은 문제이다. 오일러가 해당 문제를 풀며 그래프 이론이 탄생했다고 알려져 있다. 고등학교 자료시간에 발표하며 다루었던 내용[1]이다. 교수님께서 이 문제를 어디서 들어보았냐고 물으셨다. 나는 수학 교과서에서 보았다고 말했다. '재미있는 수학 이야기가 있어요!'와 같은 맥락에서 나왔었는데, 고등학교 교육 과정에 포함 되어 있다는 말로 이해하신 듯 하다...
2. 3개의 기둥에 크기가 각기 다른 원반을 옮기는 문제이다. 고등학교 1학년, 함지연 선생님의 플밍 강의에서 재귀로 해결했던 것이 기억난다. 그 때가 그립습니다. :)

1.4 배울 내용

논리와 명제	수학적 귀납법	집합, 데카르트 곱, 함수
재귀	행렬	Counting(세기), 순열, 조합
알고리즘	확률, 베이즈 정리	함수 복잡도, 알고리즘 복잡도
점화 관계 및 응용	정수, 소수, 해시, 암호	그래프, 트리

고백컨대 암호를 제외하면 모두 한번쯤은 배웠던 내용이다. 가장 기대되는 내용이라면 집합 → 데카르트 곱 → 함수로 이어지는 부분인데, 이 개념이 인공지능부터 양자 컴퓨팅, (특히) data base 등 방대한 분야를 망라하고 표현법에 중요한 역할을 했다고 믿기 때문이다.

[1] 발표 자료: https://www.siwonsw.com/files/paper/graph/graph.pdf

THEME 2

논리

2.1 정의

DEFINITION 2.1 (논리)**.** 논리[logic]란 명제[proposition, statements]의 진리값[truth value]을 결정하는 학문이다.

NOTE 2.1 (proposition과 statements의 차이)**.** 전지전능하신 llm에게 물어 보았다:

"Proposition"과 "statement"는 수학, 논리학, 철학 등에서 비슷하게 사용되지만, 미묘한 차이가 있습니다.

- **Proposition (명제):** 정의: 어떤 진리값을 갖는 의미 있는 문장을 의미합니다. 즉, 참(True) 또는 거짓(False)으로 평가할 수 있는 문장입니다.
  예시: "2는 짝수이다."는 참인 명제이고, "3은 짝수이다."는 거짓인 명제입니다.
  특징: 명제는 논리적 사고에서 중요한 역할을 하며, 논리 연산자와 결합하여 복잡한 논리 구조를 만들 수 있습니다.
- **Statement (진술):** 정의: 진술은 어떤 주장이나 설명을 나타내는 문장으로, 보통 의미를 전달하는 언어적 표현입니다. 진술은 반드시 참 또는 거짓일 수 있는 것은 아니며, 논리적 문맥에서 명제와 같은 의미로 사용될 수 있습니다.
  예시: "오늘 날씨가 좋다."라는 진술은 진리값이 필요하지만, 때로는 사실 여부를 판별하기 어려울 수 있습니다.

차이점: Proposition은 반드시 참 또는 거짓일 수 있는 문장이지만, statement는 일반적인 의미에서 주장, 설명, 질문 등 모든 형태의 문장일 수 있습니다. 수학이나 논리학에서 "proposition"은 주로 참 또는 거짓으로 평가할 수 있는 내용만을 의미하며, "statement"는 일상 언어나 철학적 맥락에서 좀 더 넓은 의미로 사용될 수 있습니다.

따라서 "proposition"은 논리적 분석에서 명확히 진리값을 갖는 표현을 가리키며, "statement"는 그보다 포괄적인 의미로 사용될 수 있습니다.

2.2 논리가 컴퓨터에서 왜 중요한가?

1. 알고리즘의 특징은 명확성이다. 논리를 통해 명확성을 확보하자.
2. 컴퓨터는 논리적인 기계이다. 트렌지스터는 논리 게이트에 기반해 작동한다.
3. 논리를 활용해 더 효율적인 알고리즘 제작이 가능하다.

2.3 명제

DEFINITION 2.2 (명제). 명제proposition란 참 또는 거짓으로 평가할 수 있는 문장이다.

일반적으로 명제는 문자 p, q, r, s 등으로 표현한다. 문제 상황에서 명제를 옳게 설정하는 것은 매우 중요하며, 논리를 따지는 첫 단계이다. 명제의 진리값은 명확성을 요구하여 주관성과는 거리가 멀다. 예컨대 $1 + 2 = 3$이 명제의 예이며, 진리 값은 참이다.

우리집 고양이 구름이는 누구에 물어도 귀엽다고 말하는데, 왜 '구름이는 귀엽다'가 명제가 아니지?

— 나의 생각 중

명제의 진리값은 다음과 같이 표현한다:

true	false
T	F
1	0

$x = 2$는 명제일까? x의 값에 따라 진리 여부가 결정될 때, 우리는 이를 '명제 함수'라고 부른다.

2.4 논리 연산자

DEFINITION 2.3 (논리 연산자). 논리 연산자logical operator란 명제를 연결하여 새로운 명제를 만드는 연산자이다.

'도서관이 9시에 개장하고 해는 서쪽에서 뜬다.'와 같이 논리적 상황을 자연어로 표현하는 것은 어려운 일이다. 때문에 논리적 상황을 단순화하여 수식으로 표현할 필요가 있다.

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \oplus q$	$p \longleftrightarrow q$
T	T	F	T	T	T	F	T
T	F	F	F	T	F	T	F
F	T	T	F	T	T	T	F
F	F	T	F	F	T	F	T

Table 2.1: 대표적인 논리 연산자

table 2.1와 같이 명제의 모든 경우의 수를 모아둔 표를 진리표라고 부른다.

진리표의 논리 연산자는 각각 not, and, or, implies, xor(exclusive or), iff(if and only if)를 의미한다.

2.4.1 not

- 기호: $\neq p$
- 읽는 법: not p

2.4.2 and

- 기호: $p \wedge q$
- 읽는 법: p and q

2.4.3 or

- 기호: $p \vee q$
- 읽는 법: p or q

2.4.4 implies

- 기호: $p \rightarrow q$
- 읽는 법: p implies q

2.4.5 xor

- 기호: $p \oplus q$
- 읽는 법: p xor q

2.4.6 iff

- 기호: $p \longleftrightarrow q$
- 읽는 법: p iff q

논리 연산자에 우선 순위가 존재하지만, 경험으로 모두 알 수 있다. 종이를 아끼자.

NOTE 2.2. 왜 implies, iff라는 이름을 사용할까? LaTeX에서 `\implies`는 $\Longrightarrow$, `\iff`는 $\Longleftrightarrow$로 표현된다. 이는 수학에서 참임을 검증 가능할 때 사용되는데, 명제에서는 그저 조건을 나타낼 때 표현된다. llm이 여러가지 이야기를 해주는데, 이유를 모르겠다. 그저 엄격성이 더욱 요구되는 수학과 컴퓨터 과학을 배우는 학생의 차이라고 생각하자.

2.5 복합 명제

DEFINITION 2.4 (복합 명제). 복합 명제compound proposition란 둘 이상의 명제를 논리 연산자로 연결한 명제이다.

당연하게도 우리는 복합 명제를 다루어야 한다. 보다 복잡하고 생산성 있는 일을 하기 위해서 이다.

2.6 동치

DEFINITION 2.5 (동치). 두 명제 p와 q가 동치equivalent라는 것은 두 명제가 항상(다른 말로 모든 경우에서) 같은 진리값을 갖는다는 것이다. 다음과 같이 표시한다:

$$p \equiv q \tag{2.1}$$

물론 $\iff$ 도 사용하겠다. 비단, 수업에 맞춰 가능하면 $\equiv$를 사용한다.

2.7 드모르간 법칙

DEFINITION 2.6 (드로모르간 법칙). 다음이 성립한다:

INDEX