KAIST 전산학과 최광무

증명

1.1 증명은 무엇인가?

증명은 어떤 주장이 참임을 모든(많은) 사람들이 받아들이도록 하는 방법이다.

이 강의에서는 수학(혹은 논리)증명에 대해 다룰 것이나, 실험이나 통계를 이용한 증명방법이 과학이나 공학에서 널리 이용되고, 인문사회학이나 종교에서는 사회적인 동의나 권위 혹은 개인의 믿음 등을 통해 사실을 주장하기도 한다. 논리증명(deduction)만이 수학 증명이고, 과학 실험이나 사회학 증명은 모두 경험 증명이어서 수학증명이 아니다. 심지어 통계적 방법도 엄밀한 의미에 논리증명이라고는 볼 수 없다.

(정의 1.1) 수학증명

증명은 주장(명제, statement, proposition)이 참(true)이라는 것을 공리(axiom)에서부터 시작하여, 논리(수학)적인 연역(logical deduction) 과정을 통하여 엄밀하게(formal) 밝히는 방법이다.

(정의 1.2) 명제(statement): 참이거나 거짓인 주장

주장 중에는 참이나 거짓이 명확하지 않은 주장도 있으나, 수학에서는 이러한 주장은 탐구 대상이 아니다.

(정의 1.3) 공리(axiom): 증명하지 않는 명제

명제 중에는 증명이 어렵거나, 도리어 증명하지 않는 것이 수학적 사고의 영역을 더 넓힐 수 있을 때도 있다. 수학에서 공리는 증명하지 않는 명제이다. 예를 들어, 평행한 두 개의 직선은 만나지 않는다는 주장이 사실인 것으로 경험적으로는 보이나, 이를 바꾸면 또 다른 수학적 상상력을 제공하므로, 두 개의 평행한 직선이 한 점 혹은 무한히 많은 점에서 만난다는 주장을 새로운 공리로 하여, 다른 수학체계(비유클리드 기하학)를 만들 수 있다. 다음에 설명할 연역 증명을 생각한다면, 수학증명은 공리 위에 세워진 모래성이라고 볼 수도 있다.

(정의 1.4) 논리적 연역(logical deduction).

연역(deduction)과 귀납(induction)은 증명 혹은 생각에 두 가지 큰 방법이다. 연역 증명은 뒤에서 나올 참과 거짓만이 있는 세계에서는 매우 명백하다고 판단되는 사실(deduction rules)을 통하여 증명하므로 이론의 여지가 없는 명백한 증명 혹은 생각 방법으로 받아들인다. 반면 경험(귀납)적 증명[1]은 많은 사실에서 경험적으로 추출하는 증명방법으로 수학적인 증명으로는 받아들이지 않으나 실생활에서는 많이 받아들여지는 증명(사고) 방법이다.

(정의 1.5) 엄밀한 언어(formal language): 문법(syntax)과 의미(semantics)가 수학적으로 명백한 언어

엄밀한 언어는 인간이 사용하는 언어가 명백하지 아니하여 수학자들이 만들어낸 가상의 언어이다. 상상력이나 표현력에 제한이 있으나, 명백하므로 널리 받아들여지고 있다. 주변에서 쉽게 볼 수 있는 예는 프로그램에 사용되는 Python이나 Java 등의 프로그래밍 언어가 있다. 또한, 수학에서 사용되는 각종 기호를 사용한 식(formula)이 엄밀한 언어의 좋은 예이다. 이 강

1) 수학적 귀납법은 귀납의 모양새를 갖추었지만, 완전한 연역증명이다.

의에서는 이러한 수학적 언어를 잘 사용하는 법을 배우게 될 것이다.

(정의 1.6) 정의(definition): 수학적으로 한 엄밀한 약속
정의는 “어떤 말이나 사실의 뜻을 명백히 밝혀 규정함”이라고 네이버 국어사전에 정의되어 있다. 정의는 말을 줄이기 위하여, 수학자들이 쓰는 독특한 방법이다. 삼각형을 꼭짓점이 3개인 다각형이라고 정의하고, “꼭짓점이 3개인 다각형”이라는 말 대신에 “삼각형”이라는 말로 짧게 쓰는 것이다. “꼭짓점이 세 개인 다각형”은 짧아 보이지만 사실은 “다각형”, “꼭짓점”이라는 말(용어)들이 이미 정의되어 있다는 사실을 기억하라.

정의를 다른 말로 표현하면 이름을 짓는 일이다. 정의를 통하여 명백한 이름을 지음으로써 수학자들은 필요 없는 오해나 긴말을 줄이는 것이다. 이러한 노력은 수학 이외의 다른 학문에서도 같이 일어나며, 생물학에서 분류에 사용되는 용어나 엄밀한 법학 용어들이 정의의 좋은 예이다.

학이지명명(學而之銘名[2]; 이름 짓는 것이 공부다)이라는 공자식의 사고도 존재하지만, 명가명, 비상명(名可名, 非常名[3], 이름 지어도, 이름이 본질이 아니다)이라는 노자의 세계도 존재한다. 앞에 생각이 논리에 기초한 수학적 사고라면, 뒤의 생각은 사람을 다루는 인문학 사고이다. 앞에 생각이 논리의 세계라면 뒤의 생각은 경험의 세계이다.

논리주의	경험주의
학이지명명(學而之銘名)	명가명, 비상명(名可名, 非常名)
이름 짓는 것이 공부다	이름 지어도, 이름이 본질이 아니다
논리	경험
수학, 과학	인문학, 사회학
진화(Evolution)	혁명(Revolution)
꾸준한 노력과 훈련	한순간 오는 깨달음
격물치지(格物致知)	성의정심(誠意正心)
주이론(主理論), 퇴계 이황	주기론(主氣論), 율곡 이이
교종(教宗)	선종(禪宗)

표 1.1 두 가지 세상

모든 사람이 동의하도록 엄밀하게 하는 것은 수학의 증명이고, 인문사회학의 주장은 모든 사람이 동의하기는 어려우므로, 좀 더 많은 사람이 동의할 수 있도록 노력한다. “이름 짓는” 엄밀한 사고도 중요하지만 때로는 “지은 이름을 버리는” 열린 사고도 필요하다.

그러나 수학의 증명은 한 사람도 빠지지 않고 모든 사람이 동의하는 증명을 하여야 하므로 우선은 이름을 지어나가는(“이름 짓는 것이 공부다”) 논리의 엄밀성과 성실한 노력이 필요하다.

1.2 복합명제(compound statement) - 명제의 확장 - 복합명제에 관한 엄밀한(수학) 정의
작은 명제들이 합쳐서 큰 명제를 이루는 경우가 많다. 예를 들어 “내일 비가 오거나 바람이

2) 학이지명명은 공자 논어의 학이편을 패러디한 말이다.
3) 명가명, 비상명은 노자의 도덕경 1장에서 처음으로 나오는 도가도, 비상도(道可道, 非常道)에 이서서 나오는 말이다.

불면 학교에 안 간다."라는 큰 명제를 생각하자. 이 큰 명제는 "내일 비가 온다.", "내일 바람이 분다.", "내일 학교에 간다."라는 3개의 작은 기본명제(simple statement)로 구성되어 있고, 3개의 기본명제가 "~ 이거나", "~ 이면", "안 ~"이라는 접속어로 기본명제들을 합성하여 복합명제를 만들고 있다.

1.2.1 복합명제(compound statement)의 모양새(syntax)

"내일 비가 오거나 바람이 불면 학교에 안 간다."라는 큰 명제를 수학기호(연산자; operator)를 사용하여 이거나(∨), 이면(⇒), 아니다(¬) 등으로 정리한 후에 이를 다시 써보면,

("내일 비가 온다." ∨ "내일 바람이 분다.") ⇒ ¬("내일 학교에 간다.")

로 쓸 수 있다.

연산자를 구분해 보면, 아니다(¬)라는 연산자는 대상명제를 하나만 가지는 연산자이고, 이거나(∨), 이면(⇒)이라는 연산자는 대상명제를 두 개 가지는 연산자이다. 대상명제를 하나만 가지는 연산자를 단일(unary)연산자라고 부르고, 두 개 가지는 연산자를 이진(binary) 연산자라고 부른다.

단일 연산자인 아니다(¬)는 연산자 ¬가 대상명제 앞에 나오고, 이진 연산자인 이거나(∨)와 이면(⇒)은, 연산자 ∨와 ⇒가 대상명제 둘 사이에 나온다. 연산자가 대상명제 앞에 나오는 경우를 앞(prefix) 연산자라 부르고 연산자가 대상명제 둘 사이에 나오는 연산자를 사이(infix) 연산자라고 부른다. 연산자가 대상 명제에 뒤에 나오는 뒤(postfix) 연산자도 생각할 수 있다.

대상명제의 개수(단일, 이진)와 연산자의 위치(앞, 사이, 뒤)를 생각하여 보았다. 정의하려는 수학언어의 모양새(syntax)를 엄밀히 정의하는 좋은 방법은 문법(grammar)이다. 명제의 합성인 복합명제를 정의하는 구문법(構文法; syntactic grammar)을 보자.

(정의 1.7) 합성명제의 구문법

〈명제〉 → **T** | **F** (1.7.1)
| 〈단순명제〉 (1.7.2)
| ¬〈명제〉 (1.7.3)
| 〈명제〉 ∧ 〈명제〉 (1.7.4)
| 〈명제〉 ∨ 〈명제〉 (1.7.5)
| 〈명제〉 ⇒ 〈명제〉 (1.7.6)
| **(** 〈명제〉 **)** (1.7.7)

문법 (1.7.1)은 명제(혹은 합성면제)가 참(**T**)과 거짓(**F**) 두 개의 요소(상수; constant)로 구성되어 있음을 보인다. 문법 (1.7.2)는 단순명제도 명제임을 보인다. 문법 (1.7.3)은 명제 앞에(prefix) "아니다"(¬)라는 앞 연산자를 붙이면, 새로운 명제가 됨을 보이고, 문법 (1.7.4)~(1.7.6)는 명제 두 개 사이에 "이고"(∧), "이거나"(∨), "이면"(⇒)이라는 사이(infix) 이진 연산자를 붙이면 새로운 명제를 만듦을 보인다. 문법 (1.7.7)은 명제의 앞과 뒤에 여는 괄호(**(**)와 닫는 괄호(**)**)로 구조(structure)를 만들어 명제 혹은 복합명제 사이에 우선순위(precedence와 associativity)를 조절할 수 있음을 보인다.

문법 (1.7.1)과 (1.7.2)는 명제를 기본(basis)가 되는 요소로 정의하므로, 재귀정의(Recursive definition)의 기본(basis)이라고 부르고, 문법 (1.7.3), (1.7.4), (1.7.5), (1.7.6), (.7.7)은 명제를 써서 명제를 정의하므로 재귀정의(recursion)라고 부른다.

앞으로는 상수(**T**, **F**)와 기본명제, 복합명제를 구분하지 않고, 모두 명제라 부른다.

1.2.2 복합명제(compound statement)의 뜻(semantics)

p와 q가 명제라고 하자. 복합명제 ¬p와 p ∧ q, p ∨ q, p ⊻ q, p ⇒ q, p ⇔ q의 뜻을 아래와 같이 설명한다.

1. 복합명제 아니다(not)는 ¬연산자를 앞 연산자로 사용하고 연산의 결과는 참을 거짓으로 거짓을 참으로 바꾸어준다.
2. 복합명제 이고(and)는 사이 연산자 ∧를 사용하고, 왼쪽과 오른쪽의 명제가 모두 참일 때만 참이 되고, 나머지 경우(otherwise)는 모두 거짓이다.
3. 복합명제 이거나(or)는 사이 연산자 ∨를 사용하고, 왼쪽과 오른쪽의 명제 중에 하나 이상이 참일 때만 참이고, 모두 거짓일 경우는 거짓이다.
4. 복합명제 다른(exclusive or)은 사이 연산자 ⊻(또는 ⊕)를 사용하고, 왼쪽이나 오른쪽의 명제중 하나만이 참이고 다른 명제는 거짓일 때만 참이 되고, 왼쪽 오른쪽이 모두 참이거나 모두 거짓일 때는 거짓이다.
5. 복합명제 이면(implication, if p then q)은 사이 연산자 ⇒를 사용하고, 왼쪽 명제를 가정이라고 부르고 오른쪽 명제를 결론이라고 부른다. 가정이 참이고 결론이 거짓일 때만 거짓이고, 나머지는 모두 참이다[4].
6. 복합명제 같은(p and only if q)은 사이 연산자 ⇔를 사용하고, 가정(왼쪽)과 결론(오른쪽) 명제가 같을 때만 참이고, 가정과 결론이 다를 때는 거짓이다. 즉 가정이 참이고 결론도 참이거나 가정도 거짓이고 결론도 거짓일 때만, 즉 가정과 결론 두 개의 명제가 서로 같을 때만 참이다.

명제를 합성할 때 그 뜻(semantics)이 어떠한가를 간단하고 명백하게 밝히는 방법으로 진리표(truth table)가 쓰인다. 위에서 설명한 ¬p와 p ∧ q, p ∨ q, p ⊻ q, p ⇒ q, p ⇔ q의 뜻을 아래 진리표로 정리하였다.

p	¬p
T	**F**
F	**T**

표 1.2.1 단일연산 아니다(¬)의 진리표

4) Logic says nothing, when the hypothesis is false.

p	q	p ∧ q	p ∨ q	p ⊻ q	p ⇒ q	p ⇔ q
T	T	T	T	F	T	T
T	F	F	T	T	F	F
F	T	F	T	T	T	F
F	F	F	F	F	T	T

표 1.2.2 이진연산 이고(∧), 이거나(∨), 다른 이거나(⊻), 이면(⇒), 같은(⇔)의 진리표

1.3 연역규칙(Deduction(Inference) Rules)

1. 참 연역(Modus ponens)
   가정: p, p ⇒ q
   결론: q.
2. 거짓 연역(Modus tolens)
   가정: p ⇒ q, ¬q
   결론: ¬p.
3. 3단 논법(Syllogism)
   가정: p ⇒ q, q ⇒ r
   결론: p ⇒ r.
4. 모순법(proof by contradiction)
   가정: ¬p ⇒ 거짓
   결론: p.
5. 거짓결론
   가정: p, ¬p
   결론: 거짓.
6. 경우(Case)
   가정: p ⇒ q, ¬p ⇒ q
   결론: q.

원소가 "참"과 "거짓" 두 개인 집합 𝔹 = {참, 거짓}을 생각하자. 집합 𝔹를 "참과 거짓" 또는 Boolean이라고 부르겠다. 또한, 참과 거짓이 좀 기므로, 참 = T로 거짓 = F로 줄여서 기호(symbol)를 사용하여 정의하여 𝔹 = {T, F}로 쓰겠다.

1.3.1 연역규칙(Deduction rules)의 공리(axiom)

모든 명제는 Boolean 집합 𝔹의 원소이다.

연역규칙의 공리는 명제가 참과 거짓 두 가지 경우만 가진다는 것이다. 이 공리를 기본으로 하여 위의 여섯 가지 연역규칙을 모두 증명할 수 있다. 증명하는 방법은 명제가 T나 F 이외에는 어떤 값도 가질 수 없다는 공리와 복합명제의 의미를 정의한 진리표를 이용하여 증명할 수 있다.

1.3.2 증명의 예

1.3.2.1 직접증명(Direct proof)

명제 p ⇒ q가 참임을 증명하기 위하여 위의 여섯 가지 연역규칙 중에 첫 번째 규칙인 참 연역을 약간 변형하여 p, q이면 p ⇒ q를 이용하여 증명한다.
증명: p ⇒ q
1. p(가 참임)를 가정한다.
2. q도 참임을 증명한다.
3. 증명 끝(QED)

(예 1.1) 1 + 2 + ⋯ + 100 = 5050의 증명
증명: S = 1 + 2 + ⋯ + 100 ⇒ S = 5050.
1. S = 1 + 2 + ⋯ + 100라고 가정하자.
2. S = 5050 ???
3. 증명 끝(S = 1 + 2 + ⋯ + 100 ⇒ S = 5050).

??? 부분의 증명이 더해져야 한다.
증명: S = 1 + 2 + ⋯ + 100 ⇒ S = 5050.
1. S = 1 + 2 + ⋯ + 100라고 가정하자.
2. S = 5050
    1. S = 100 + 99 + ⋯ + 1 +에 관한 배분법칙(associativity)
    2. 2S = 101 + 101 + ⋯ + 101 식1과 식2.1을 항끼리 더한다.
    3. 2S = 101 * 100 = 1,0100 101이 100번 더하여진다.
    4. 증명 끝(즉 S = 5050이다) 등호의 양변을 2로 나눈다.
3. 증명 끝(S = 1 + 2 + ⋯ + 100 ⇒ S = 5050).

(예 1.2) DeMorgan의 법칙이라는 집합의 연산에 관한 규칙의 일부의 증명. 즉 x가 집합 A와 B의 교집합(intersection)의 보집합(complement)의 원소이면, x는 A와 B의 합집합(union)의 보집합의 원소이다.
증명: $x \in \overline{A} \cap \overline{B} \Rightarrow x \in \overline{A \cup B}$
1. $x \in \overline{A} \cap \overline{B}$라고 가정하자.
2. $x \in \overline{A \cup B}$
    1. $x \in \overline{A} \wedge x \in \overline{B}$ $\cap$의 정의
    2. $x \notin A \wedge x \notin B$ $\overline{A}$와 $\overline{B}$의 정의
    3. $x \notin A \cup B$ $\cup$의 정의
    4. $x \in \overline{A \cup B}$ $\overline{A \cup B}$의 정의
    5. 증명 끝($x \in \overline{A \cup B}$).
3. 증명 끝($x \in \overline{A} \cap \overline{B} \Rightarrow x \in \overline{A \cup B}$).

1.3.2.2 모순증명(Proof by contradiction)
p를 증명하기 위하여, ¬p ⇒ 거짓(contradiction)임을 증명하면 된다.

증명: p

1. $\neg$p $\Rightarrow$ 거짓
2. 증명 끝(p)

위 증명의 변형은 아래와 같다.

증명: p
1. $\neg$p $\Rightarrow$ (q $\wedge$ $\neg$q)
2. 증명 끝(p)

증명: p
1. $\neg$p $\Rightarrow$ q
2. $\neg$q
3. 증명 끝(p)

(예 1.3) $\sqrt{2}$는 무리수다.
증명: $\sqrt{2}$는 무리수다.
1. $\sqrt{2}$는 유리수 $\Rightarrow$ 거짓
2. 증명 끝($\sqrt{2}$는 무리수이다).

$\sqrt{2}$가 유리수임을 가정한 후에 $\sqrt{2}$가 유리수이면 거짓($\sqrt{2}$는 유리수 $\Rightarrow$ 거짓)임을 모순법으로 증명하여 보자.

증명: $\sqrt{2}$는 무리수다.
1. $\sqrt{2}$는 유리수 $\Rightarrow$ 거짓
    1. $\sqrt{2}$는 유리수라고 가정하자.
    2. 거짓

1. $\sqrt{2} = \frac{a}{b}$인 $a, b \in N^+$중에 가장 작은 수라고 가정.	유리수의 정의
2. $2b^2 = a^2$	산술
3. a^2는 짝수이다.	짝수의 정의
4. a도 짝수이다.	짝수의 성질
5. $a = 2c$인 $c \in N^+$를 생각하자.	짝수의 성질
6. $2b^2 = a^2 = 4c^2$	식 1.2.2와 1.2.5
7. $b^2 = 2c^2$	산술
8. b^2도 짝수이다.	짝수의 정의
9. b도 짝수이다.	짝수의 성질
10. $\sqrt{2} = \frac{a}{b}$인 $a, b \in N^+$중에 가장 작은 수가 아니다.	1.2.4와 1.2.9
11. 증명 끝(거짓)	1.2.1과 1.2.10

    3. 증명 끝($\sqrt{2}$는 유리수 $\Rightarrow$ 거짓) 1.1과 1.2이면 (1.1 $\Rightarrow$ 1.2)

2. 증명 끝($\sqrt{2}$는 무리수다). 유리수가 아니면 무리수이다.

1.3.2.3 경우에 따른 증명(Proof by cases)