

기초정수론 (Elementary Number Theory)

2023년 5월 24일

시니어 수학교실 (Math for Seniors)

유튜브: <https://www.YouTube.com/@mathforseniors>

저자: JB

이메일: mathforseniors@gmail.com

Contents

1 강의소개와 사전지식	2
1.1 이상적인 수강생 모델	2
1.2 주의사항	3
1.3 강의 주제, 등록, 일정	3
1.4 사전지식	4
1.4.1 집합(Sets)	4
1.4.2 논리(Logic)와 명제(Proposition)	5
2 수체계(Numbers)와 방정식(Equations)	8
2.1 자연수(Natural Numbers)와 연산들(Operations)	8
2.2 자연수(Natural Numbers)와 방정식(Equations)	9
2.3 정수(Integers)와 방정식(Equations)	11
2.4 유리수(Rational Numbers)와 방정식(Equations)	11
2.5 실수(Real Numbers)와 방정식(Equations)	11
2.6 복소수(Complex Numbers)	11

Chapter 1

강의소개와 사전지식

본 강좌는 정수론의 기초와 그를 이용한 작은 개인/그룹과제 해결을 통해 수학에 대한 관심과 소양을 얻기위해 만들어졌습니다. 더 중요한 목적은 실생활에서의 응용외에 하나의 철학으로 삶의 의지와 의미를 찾기위함에 있습니다.

1.1 이상적인 수강생 모델

강좌는 아래와 같은 조건에 해당하시는 수강생들이 공부하기에 최적화되어 있습니다.

- 은퇴를 하셨거나 여유시간이 많으나 의미있는 활동을 찾기힘드신 분 중
지적 성취감을 느끼고 싶으신 분.
- 평소 수학/과학/공학에 관심이 많으신 분.
- 자극적이고 단편적인 뉴스, 유튜브등의 미디어에 중독되어 이를 극복하고자
하시는 분.
- 논리적인 사고력을 키워 토론에서 자신의 의견을 합리적으로 표현하고자
하시는 분.
- 수학을 통해 세상과 신을 이해하고 싶으신 분.

1.2 주의사항

본 강좌에 관한 주의사항은 아래와 같습니다.

- 정수론의 소양과 관련이 적은 주제에 대해 지나치게 염밀한 논리가 필요한 부분은 효과적인 강의를 위해 가급적 피하겠습니다.
- 수학용어들은 영어번역을 같이 표기하겠습니다. 우리가 쓰는 한글 수학용어는 가끔 혼동을 일으키곤 합니다. 일례로 ‘소수(小數)’와 ‘소수(素數)’는 동음이의어로 한글로만 썼을때 상당한 혼란을 가져올 수 있습니다. 그리고, 차후 논문이나 원서를 일으실때 미리 용어를 영어로 알아두시는것도 좋을꺼라 생각합니다.
- 마지막으로 가장 중요한 주의사항은 **최대한 본인들이 직접 문제를 해결하려고 노력해야합니다.** 이것은 절대적인 요소입니다. 수학은 미디어에서 보여지는 것과 달리 실제로는 수많은 시행착오와 오랜 시간의 고민을 통해서 답을 찾을 수 있습니다. 만약 참을 수 없을 정도로 오래 노력해도 답을 얻을 수 없을때 본 강사나 동료들과의 대화에서 힌트를 얻기를 바랍니다.

1.3 강의 주제, 등록, 일정

본 강좌에서 공부할 토픽들을 다음과 같습니다.

- 수체계: 정수(Integers), 유리수(Rational numbers), 실수(Real Numbers), 복소수(Complex Numbers), 대수적 수(Algebraic Numbers), 초월수(Transcendental Numbers) 등.
- 연산(Operations), 모듈러 연산(Modular Arithmetic)
- 방정식(Equations), 함수(Functions)

등록한 수강생 수가 10명이상이 되면 좀더 즉각적인 소통을 위해 유튜브 온라인 강의도 계획하고 있습니다. 등록은 강사의 이메일 mathforseniors@gmail.com로 성명 또는 원하시는 예명(일례로 유튜브에서 사용하는 이름), 이메일, 짧은 자기소개와 수강이유를 보내주시면 되겠습니다.

첫 10강의는 차후 공개될 개인/그룹 프로젝트 문제들을 해결하기 위해 필요한 정수론의 기본 정의(Definition)/용어(Terminology)/정리(Lemma, Proposition, Theorem)들을 공부하겠습니다.

수강생에게 부여될 프로젝트 문제들의 답은 정해진 날짜전에 강사의 이메일이나 저장소에 제출하시면 되겠습니다. 제출된 모든 답은 강사의 리뷰와 함께 모든 수강생들이 열람 가능하도록 하겠습니다. 수강생 각자 수학자라고 생각하고 자신의 답에 자부심을 가지게 되었으면 합니다.

1.4 사전지식

본 사전지식은 집합과 논리 그리고 명제에 대해 공부해보겠습니다. 우리는 정수론을 위해 힘을 아껴야하니 이 부분에 대해서 지나친 힘을 낭비하지 마시길 바랍니다. 바로 이해가 안되시다면 그 부분은 읽고 지나가셔도 됩니다.

1.4.1 집합(Sets)

정의 1.4.1. 집합(Set)은 정의할 수 있는 서로 다른 객체들의 모임이다. 주어진 집합 S 안의 각 객체 a 를 S 의 원소(Element)라고 부르고 $a \in S$ 라고 쓰고, 만약 어떤 객체 b 가 집합 S 에 속해 있지 않다면 $b \notin S$ 라고 씁니다.

어떤 객체도 없는 집합도 정의할 수 있습니다. 마치 정수에서 0처럼 말이죠. 이 특별한 집합을 공집합(Empty set)이라고 부르고 기호로는 $\{\}$ 또는 \emptyset 을 씁니다.

집합을 수학적으로 표기할 때의 규정은 다음과 같습니다. 만약 집합 A 는 원소 a, b, c 를 갖는다고 하면 $A = \{a, b, c\}$ 라고 씁니다. 자연수(Natural numbers)는 1부터 1씩 더해지는 원소들의 집합이라고 정의되고 간단히 \mathbb{N} 이라고 쓰는데 이것을 위와 같이 모든 원소들을 다 나열해서 쓰기는 불가능하죠? 이렇듯 원소들이 자명할 때는 그냥 $\{1, 2, \dots\}$ 라고 쓰기도 합니다.

자연수 중 모든 짝수들의 집합을 어떻게 표현하면 좋을까요? 물론 $\{2, 4, 8, \dots\}$ 이라고 쓸 수 있지만 $\{2a \mid a \in \mathbb{N}\}$ 라고 쓸 수도 있습니다. 다시 말해, 자연수 집합 \mathbb{N} 의 각 원소 a 에 2를 곱해서 만들어지는 수들의 집합으로 표현할 수 있습니다. 그럼, 자연수 중 모든 홀수들의 집합은 어떻게 표현할 수 있을까요? 각자 생각해봅시다.

위에서 봤듯이, 모든 집합은 원소의 갯수를 셀 수 있는 집합과 그렇지 않은 집합으로 분류됩니다. 전자를 유한집합(Finite set), 후자를 무한집합(Infinite set)이라고 부릅니다. 일례로 \emptyset 은 원소의 갯수가 0인 유한집합, 자연수 집합 \mathbb{N} 은 무한집합입니다.

정의 1.4.2. 집합 A 의 모든 원소가 집합 B 의 원소일 때 A 는 B 의 부분집합(Sub-set)이라고 하고, $A \subset B$ 라고 씁니다.

두 집합 A 와 B 가 같다라는 것을 수학적으로 어떻게 정의할 수 있을까요? 한 방법은 A 에서 어떤 원소 a 를 꺼내서 B 에서 a 를 찾아서 제거하는 과정을 A 안의 모든 원소에 대해 거치면 결국 B 는 공집합이 된다로 할 수 있겠죠. 괜찮은 방법인데 좀 더 생각해보면 같은 집합에 대한 정의를 간단히 부분집합으로도 할 수 있습니다. 즉 $A \subset B$ 그리고 $B \subset A$ 일 때 $A = B$ 라고 정의할 수도 있겠죠?

집합들 사이에는 연산도 존재합니다. 두 집합 A 와 B 를 이용해 어떻게 다른 집합을 만들어낼 수 있을까요? 우리가 자주 쓰는 연산들은 다음과 같습니다.

- 곱집합(Intersection): $A \cap B := \{x \mid x \in A \text{ 그리고 } x \in B\}$
- 합집합(Union): $A \cup B := \{x \mid x \in A \text{ 또는 } x \in B\}$
- 여집합(Difference set): $B - A := \{x \mid x \in B \text{ 그리고 } x \notin A\}$

예를 들어, $B = \{0, 1, 2\}$ 이고 $A = \{1, 3\}$ 이라면 $A \cap B = \{1\}$, $A \cup B = \{0, 1, 2, 3\}$, $B - A = \{0, 2\}$ 그리고 $A - B = \{3\}$ 입니다. 여기서 한 가지 재밌는 점은 $A \cap B = B \cap A$ 이고 $A \cup B = B \cup A$ 이지만 $A - B$ 와 $B - A$ 는 항상 같지는 않다는 점이죠. 마치 정수들의 덧셈은 서로 항들을 교환 가능하지만 뺄셈은 안 되는 것과 마찬가지로요.

다음에는 논리와 명제에 대해 알아보도록 하겠습니다.

1.4.2 논리(Logic)와 명제(Proposition)

정의 1.4.3. 명제(Proposition)란 참(True)이거나 거짓(False)이면서 동시에 참과 거짓이 아닌 주장(Statements/Assertions)입니다.

여기서 주의할 점은 수학에서 명제란 동시에 참이거나 거짓이 될수는 없습니다. 일례로, "내가 지금하는 주장은 거짓이다."가 명제 P 라고 합시다. P 가 참이라면 내가 지금하는 주장 P 는 거짓이므로 서로 모순이겠죠? 반대로 P 가 거짓이라면 바로 전 논리를 이용해 P 가 참이 되는 모순이 발생합니다. 그리고, 아무 문장이나 다 명제가 되는것은 아닙니다. 예를 들어, "이리와봐.", "오늘 점심엔 무엇을 먹을까?" 같은 참 또는 거짓이라고 할 수 없으므로 명제가 아닙니다.

"그리고(And)", "또는(Or)" 등의 기본적인 논리연산들은 우리가 살아오면서 다 경험적으로 체득하셨을테니 정수론에서 필요한 몇가지 논리와 증명방법으로 넘어가겠습니다.

먼저, 수학에서 자주 나오는 논증(Arguments) 중에 두 명제 P 와 Q 사이에

$$P \text{ 이면 } Q.$$

라는것이 있습니다. 예를 들면 명제

자연수 a 에 대해 $a + 1$ 이 짝수라면 a 는 홀수다.

를 생각해봅시다. 수학에서는 이것을 다음과같이 표기합니다:

$$P \implies Q.$$

위의 명제가 참인지 아닌지, 즉 참임 증명하는데 여러 방법들이 있을텐데, 수학에서 종종쓰는 방법은 " $P \implies Q$."를 증명하는 대신 " Q 가 아니면 P 가 아니라."를 증명하는 것입니다. 대우(Contra-positive)라고 불리는데요. 위의 예에서 a 가 홀수가 아니라면 (즉 짝수라면) 거기에 1을 더한 수 $a + 1$ 은 홀수가 되므로(즉 짝수가 아님으로), 증명을 할 수 있게 되는 것이죠. 결국, 두 명제 " $P \implies Q$ " 와 " P 가 아님 $\implies Q$ 가 아니다"는 서로 논리적으로 같은 말입니다. 우리는 이럴때 다음과같이 표기합니다:

$$"P \implies Q" \iff "P\text{가 아님} \implies Q\text{가 아니다}."$$

주의사항: $A = B$ 와 $A \iff B$ 는 참이거나 거짓일 수는 있지만 수학에서 같은 의미를 갖고 있지는 않습니다.

또 비슷한 방법은 모순을 이끌어내어 증명하는 방법이 있는데, 유명한 예로 "소수(Prime numbers)는 무한히 많다."가 있습니다. 차후에 배우겠지만 "소수는

"유한하다"고 가정하고 어떤 모순을 이끌어내어 그 가정 "소수는 유한하다"는 거짓이므로, "소수는 무한하다"고 증명을 이끌어낼 수 있는 것이죠. 이 방법은 상당히 효과적이고 간단하죠.

다른 효과적인 증명방법으로 수학적 귀납법(Induction)이라는 것이 있습니다. 이 방법은 무한집합 자연수를 논리적으로 이용하는 것인데요. 간단한 적용사례로 다음과 같은 명제를 생각해 봅시다.

$$\text{"모든 자연수 } n\text{에 대해 } 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}\text{이다."}$$

먼저 $n = 1$ 일 때를 증명합니다: $1 = 1(1+1)/2 = 2/2 = 1$. 다음, 어떤 자연수 n 일 때 위의 명제가 참이라고 가정하고 (즉 $1 + 2 + 3 + \dots + n = n(n+1)/2$), $n + 1$ 일 때를 증명합니다:

$$1 + 2 + 3 + \dots + n + (n+1) = \frac{n(n+1)}{2} + n + 1 = \frac{(n+1)(n+2)}{2}.$$

직관적으로 첫번째 $n = 1$ 일 때 참임을 증명하고 그뒤에 커지는 자연수는 연쇄적으로 자동으로 다 참임이 증명되므로 모든 자연수에서 참임이 증명된다는 아이디어입니다.

마지막으로 어떤 명제가 거짓임을 증명하는 것도 자주 등장하는데요. 보통 처음으로 시도해볼 수 있는 방법은 반례(Counter-example)를 찾아보는 것입니다. 반례를 하나만 찾아도 되겠죠? 보통 수학자들은 정리를 증명하기 어렵지만 가치가 있는 주장을 추측(Conjecture) 또는 가설(Hypothesis)이라고 발표하는데요. 추측이 나오면 제일 먼저 해보는 것은 실제로 추측이 참인지 거짓인지 예를 많이 찾아보는 것입니다. 그 중에 추측에 반하는 반례를 찾으면 그 즉시 추측은 쓸모 없게 되버리죠.

여기까지 정수론을 위한 필수적인 소양이었구요. 다음 챕터부터는 본격적으로 정수론에 대해 공부하겠습니다.

Chapter 2

수체계(Numbers)와 방정식 (Equations)

2.1 자연수(Natural Numbers)와 연산들(Operations)

먼저, 앞으로 \mathbb{N} 라고 하면 집합으로서의 자연수 $\{0, 1, 2, 3, \dots\}$ 뿐만 아니라 덧셈(Addition)과 곱셈(Multiplication)에 대한 연산을 가지는 집합으로 쓰겠습니다.

덧셈은 문자나 숫자를 사용하는 경우 모두 '+'로 표기합니다. 예를들면, $1 + 2$, $a + b$. 곱셈은 ' \cdot '을 쓰고, 종종 생략하기도 합니다. 예를들면, $1 \cdot 2$, ab , $a \cdot a = a^2$, $2 \cdot a = 2a$.

참고 2.1.1. \mathbb{N} 에서 어떤 두 자연수 a, b 를 더하거나 곱하면 그 결과가 자연수임을 알고있죠? 다시말해 그 결과가 집합 \mathbb{N} 밖으로 빠져나가는 경우는 없다와 같은 말이고, 우린 이럴때 \mathbb{N} 은 덧셈과 곱셈에 대해 닫혀있다.'라고 합니다. 영어로는 ' \mathbb{N} is closed under the addition and multiplication.'

참고 2.1.2. \mathbb{N} 에서 덧셈과 곱셈은 교환법칙(Commutativity)을 만족합니다. 수학적으로 보면 모든 $a, b \in \mathbb{N}$ 에 대해

$$a + b = b + a \text{ 그리고 } ab = ba.$$

즉 두 자연수 사이의 연산순서는 상관없다, 다른말로 연산결과는 연산순서에 불변Invariant)이다.

참고 2.1.3. 또 다른 법칙으로는 3개 이상의 자연수 사이의 덧셈과 곱셈에서 결합법칙(Associativity)을 만족합니다. 수학적으로 모든 $a, b, c \in \mathbb{N}$ 에 대해

$$(a + b) + c = a + (b + c) \text{ 그리고 } a(bc) = (ab)c.$$

수학에서 괄호 (), {}, []는 우선순위를 나타내는 특별한 기호입니다. $(a + b) + c$ 의 연산순서는 ()안의 연산($a + b$)을 먼저한 뒤 그 결과를 c 와 더하는 것입니다. 결합법칙은 3개 이상의 자연수 사이에서 괄호를 앞에 두개에 쓰던 뒤에 두개에 쓰던 같은 연산값을 갖는 것을 말합니다.

참고 2.1.4. 마지막으로 3개 이상의 자연수 사이의 덧셈과 곱셈이 같이쓰일때 분배법칙(Distributivity)을 만족합니다. 수학적으로 모든 $a, b, c \in \mathbb{N}$ 에 대해

$$(a + b)c = ac + bc \text{ 그리고 } a(b + c) = ab + ac.$$

주의할 점은 일반적으로 $(a + b)c \neq a(b + c)$ 입니다. 예를들어

$$9 = 3 \cdot 3 = (1 + 2)3 = 1 \cdot 3 + 2 \cdot 3 = 3 + 6 = 9$$

이지만

$$(1 + 2)3 = 3 \cdot 3 = 9 \neq 5 = 1 \cdot 5 = 1(2 + 3)$$

TODOTODO: Inequalities

2.2 자연수(Natural Numbers)와 방정식(Equations)

\mathbb{N} 의 어떤 상수(Constants), 변수(Variables)와 덧셈/곱셈으로 "잘" 구성된 표현을 생각해봅시다. 여기서 상수란 어떤 고정된 수를 표현한 문자, 변수란 고정이 되지 않은 또는 어떤 수인지 아직 모르는 수를 대표하는 문자입니다. 일례로 서울에 사는 특정하지 않은 한 사람을 표현하고 싶을때 문자로 대치해서 변수로 쓸 수 있겠죠? 미지수를 x 라고 하고 $2x + 1$ 은 잘 구성된 표현이고, $1++=x$ 이건 아닙니다. 왜냐하면, 표현 ' $++=$ '는 어떤 의미를 가지는지 정의되지 않았기 때문이죠.

수학에서 두 표현 A, B 가 있다고 하면 방정식은 $A = B$ 라고 쓰는 식(Formula)입니다. 일례로, $2x + 1 = 13$, $5x^3 + 2x + 1 = 0$. 물론 여러개의 변수를 쓸 수 있죠. 중요한 점은 방정식을 정의할때 변수와 고정된 상수(Constants)는 다른

개념이기 때문에 어떤 문자가 상수인지 변수를 잘 특정해야합니다. 예를 들어, 변수 $x, y \in \mathbb{N}$ 와 상수 $c \in \mathbb{N}$ 를 가지는 방정식 $2x + 3xy^2 + c = 0$.

어떤 주어진 방정식을 푼다라는 말은 왼쪽과 오른쪽 표현들이 같은 값을 갖게 하는 변수(들)의 특정값을 찾는다와 같은 말입니다. 다시 말하면, 해(Solutions)를 찾는다와도 같은 말입니다. 실제로 쉬운 방정식을 하나 풀어보죠. 변수 $x \in \mathbb{N}$ 를 갖는

$$3x + 1 = 7$$

을 생각해 봅시다. 뺄셈과 나눗셈을 알고 계신 분들은 쉽게 풀 수 있겠지만 아직 정의하지 않은 연산들이기 때문에 우리는 각 특정값들을 대입해서 방정식을 만족하는지 찾는 방법을 쓸 수 있겠죠. 0부터 x 에 대입하면 즉

- $x = 0$ 이면 방정식은 $3 \cdot 0 + 1 = 0 + 1 = 1 \neq 7$
- $x = 1$ 이면 $3 \cdot 1 + 1 = 3 + 1 = 4 \neq 7$
- $x = 2$ 이면 $3 \cdot 2 + 1 = 6 + 1 = 7$

별써 방정식을 만족하는 x 의 자연수 하나를 찾았네요.

연습문제 2.2.1. 자연수 $x = 2$ 외에 위의 방정식을 만족하는 다른 자연수가 또 있을까요? 있다면 찾아보시고, 없다면 왜 없는지 증명할 수 있을까요?

그럼 다른 방정식도 풀어봅시다: 변수 $x \in \mathbb{N}$ 를 갖는

$$3x + 1 = 5.$$

위와 같은 방법으로 풀어보죠.

- $x = 0$ 이면 방정식은 $3 \cdot 0 + 1 = 0 + 1 = 1 \neq 5$
- $x = 1$ 이면 $3 \cdot 1 + 1 = 3 + 1 = 4 \neq 5$
- $x = 2$ 이면 $3 \cdot 2 + 1 = 6 + 1 = 7 \neq 5$
- $x = 3$ 이면 $3 \cdot 3 + 1 = 9 + 1 = 10 \neq 5$

계속 다른 x 값을 대입해 계속 시도해야 할까요? 왜 더 이상 할 필요가 없을까요? 자연수에 대해 x 가 커지면 $3x$ 도 커지고 그러면 $3x + 1$ 도 커지게 되겠고, 그럼

x 가 2 이상이면 즉 $x \geq 2$ 이면 $3x + 1 \geq 7$ 이므로 5와 같아질 수 없죠. 그러므로, 위의 방정식을 만족하는 자연수 \mathbb{N} 에서의 x 값 즉 해(Solution)는 없죠.

참고 2.2.2. 앞으로는 임의의 수체계에서 특별한 언급이 없을 경우 x, y, z 는 변수를 뜻하는 문자들로 사용하겠습니다.

연습문제 2.2.3. 변수 x 외에 위의 방정식을 만족하는 다른 자연수가 또 있을까요? 있다면 찾아보시고, 없다면 왜 없는지 증명할 수 있을까요?

TODOTODO: Identity, Inverse, Closureness for addition and multiplication, Inequalities

이제 방정식과 수체계사이의 관계에 대한 역사에 대해 알아보죠. 자연수 \mathbb{N} 에 특별한 수 0이 왜 포함됐을까요? 먼저 0이 왜 특별할까요? 그건 덧셈과 관련이 있는데, 다음과 같은 방정식을 생각해봅시다. 변수 $x \in \mathbb{N}$ 와 상수 $a \in \mathbb{N}$ 를 가지는 방정식

$$x + 1 = a.$$

2.3 정수(Integers)와 방정식(Equations)

2.4 유리수(Rational Numbers)와 방정식(Equations)

2.5 실수(Real Numbers)와 방정식(Equations)

2.6 복소수(Complex Numbers)

Bibliography

- [AF] AMS Author FAQ, <http://www.ams.org/authors/author-faq>
- [MDF] The `mdframed` package, Marco Daniel and Elke Schubert, 2013/07/01, v1.9b, <http://mirror.ctan.org/macros/latex/contrib/mdframed>
- [NDS] The `needspace` package, Peter Wilson, 2010/09/12, v1.3d, <http://mirror.ctan.org/macros/latex/contrib/needspace>
- [THT] `Thmtools` Users' Guide, Ulrich M. Schwarz, 2014/04/21 v66, <http://mirror.ctan.org/macros/latex/exptl/thmtools>