이산수학

휴먼지능정보공학전공

학습 내용

- 증명
  - 수학적귀납법
  - 직접증명
  - 간접증명
  - 재귀법

수학적 귀납법

- 정의와 공리
  - 공리는 별도의 증명 없이 참(T)으로 이용되는 명제임
    - 집합 중에 어떤 것 포함하지 않은 집합이 공집합으로 존재함
    - 임의의 자연수 n에 대해 n+1이 존재함
  - 정의는 논의의 대상을 보편화하기 위해 사용하는 용어 또는 기호의 의미를 확실하게 규정한 문장이나 식
    - 네 변의 길이가 같은 사각형을 정사각형이라고 함
    - 한 내각의 크기가 직각인 삼각형을 직감삼각형이라고 함
    - 명제는 참이나 거짓으로 판별할 수 있는 문장이나 식

수학적 귀납법

- 정리와 증명
  - 정리는 공리와 정의를 통해 참으로 확인된 명제임
    - 피타고라스 정리: 직각삼각형은 (빗변의 길이2) = (밑변의 길이2) + (높이의 길이2)이다.
    - 나머지 정리: 다항식 f(x)를 몫으로 나누어 몫보다 한 차수 이하의 나머지를 구하는 정리이다.
  - 증명은 하나의 명제가 참임을 확인하는 과정으로 논리적 법칙으로 주어진 가정으로부터 결론을 유도해가는 추론의 방법으로 명제나 논증이 타당한지를 입증하는 것이라고 할 수 있음
    - 증명 방법에는 직접증명법, 간접증명법, 수학적 귀납법이 있음

수학적 귀납법

- 수학적 귀납법
  - 자연수 n에 관한 명제 p(n)이 모든 자연수 n에 대해 만족하는 것을 다음 세 단계의 과정으로 증명하는 방법
    - 기본가정(기초과정)
      - P (논의영역의 초기값)가 참(T)임을 증명한다.
    - 귀납가정
      - 임의의 k에 대해 p(k)가 참이라고 가정한다.
    - 귀납단계
      - 기본가정과 귀납가정을 이용해 k+1에 대해 p(k + 1)이 참임을 증명

수학적 귀납법

문제

- $n \geq 1$ 인 자연수에 대해 다음 식이 성립함을 수학적 귀납법으로 증명하세요
- $1+2+\cdots+n = (n(n+1))/2$

- p(n) = 1+2+…+ n = $(n(n+1))/2$ 일 때,

기본가정) 이 연산식은 n≥1 인 자연수에 대한 식이므로 초깃값이 1임을 알 수 있다.

∴ p(1) = 1 = (1(1+1))/2 = 1 로 성립한다.

귀납가정) p(k) = 1+2+…+ k = $(k(k+1))/2$가 성립한다고 가정한다.

귀납단계) 위의 두 가정을 이용해 p(k+1) = 1+2+…+ k + (k + 1) = $((k+1)\{(k+1)+1\})/2$가 성립하는지 증명한다.

수학적 귀납법

문제

- $n \geq 1$ 인 자연수에 대해 다음 식이 성립함을 수학적 귀납법으로 증명하세요
- $1+2+\cdots+ n = (n(n+1))/2$

- p(k+1) = 1+2+…+ k + (k + 1) = $((k+1)\{(k+1)+1\})/2$ = $(k+1)(k+2)/2$

귀납과정에서 1+2+…+ k = = $k(k+1)/2$이 성립한다고 가정했으므로,

p(k+1) = 1+2+…+ k + (k + 1)

= $k(k+1)/2$ + (k + 1)

= $((k+1)+2(k+1))/2$

= $(k^2+k+2k+2)/2$ = $(k^2+3k+2)/2$

= $((k+1)+(k+2))/2$

∴ p(k+1) = 1+2+…+ k + (k + 1) = $((k+1)\{(k+1)+1\})/2$

∴ 기본가정과 귀납가정에 의해 p(k+1) 은 참이다.

수학적 귀납법

문제

- $n \geq 5$인 자연수에 대해 $2^n > n^2$ 성립함을 수학적 귀납법으로 증명하세요

- $p(n) = 1+2+\ldots+ n = (n(n+1))/2$ 일 때,
  $p(n) : 2^n > n^2$ 일 때,
  기본가정) $n \geq 5$인 자연수에 대한 식이므로 초깃값은 5임을 알 수 있다.
  $\therefore p(5) : 2^5 = 32 > 25 = 5^2$ 이므로 성립한다.
  귀납가정) $p(k) : 2^k > k^2$ 이 성립한다고 가정한다.

수학적 귀납법

문제

- $n \geq 5$인 자연수에 대해 $2^n > n^2$ 성립함을 수학적 귀납법으로 증명하세요

- 귀납단계) $p(k + 1) : 2^{(k+1)} > (k + 1)^2$ 이 성립함을 증명한다. ①
  $2^{(k+1)}$은 2×2^k이므로 귀납가정의 좌변에 2를 곱한 것과 같은 형태다.
  귀납가정의 양변에 2를 곱하면, $2 \times 2^k > 2k^2$. ②
  p(k) 는 $n \geq 5$인 자연수에 대한 명제이므로 ①의 $(k + 1)^2$과 ②의 $2k^2$ 비교하면,
  $2k^2 > (k + 1)^2 : 2k^2 - (k + 1)^2 = 2k^2 - (k^2+2k+1) = 2k^2 - k^2 - 2k - 1$
  $= k^2 - 2k - 1 = (k - 1)^2 - 2 > 0 \quad (\because k \geq 5)$
  $\therefore 2^{(k+1)} > 2k^2 > (k + 1)^2$
  $\therefore k \geq 5$인 자연수에 대해 $p(k + 1) : 2^{(k+1)} > (k + 1)^2$ 은 참이다.

수학적 귀납법

문제
▪ $n \geq 3$인 자연수에 대해 $n^2 > 2n+1$이 성립함을 수학적 귀납법으로 증명하세요

- $p(n) : n^2 > 2n+1$ 일 때,
  기본가정) $n \geq 3$인 자연수에 대한 식이므로 초깃값은 3임을 알 수 있다.
  $\therefore p(3) : 3^2 = 9 > 2 \cdot 3 + 1 = 7$이므로 성립한다.

수학적 귀납법

문제

- $n \geq 3$인 자연수에 대해 $n^2 = 2n+1$이 성립함을 수학적 귀납법으로 증명하세요

- 귀납가정) $p(k)$: $k^2 > 2k + 1$ 이 성립한다고 가정한다.
  귀납단계) $p(k + 1)$: $(k + 1)^2 > 2(k + 1)+1$ 이 참임을 증명한다.
  $(k + 1)^2 = k^2 + 2k + 1 > 2(k + 1)+1 = 2k + 3$ ①
  귀납가정에서 $k^2 > 2k + 1$이라고 했으므로 귀납가정의 양변에 $2k + 1$을 더하면
  $k^2 + 2k + 1 > 2k + 1 + 2k + 1$ $(= 4k + 2)$ .... ②
  $p(k)$ 는 $k \geq 3$인 자연수에 대해서만 성립하므로 ②의 $4k+2$는 ①의 $2k+3$보다 크다.
  ($\because$ $4x3+2=14$, $2x3+3=9$). 따라서 $k^2 > 2k + 1 > 4k+2 > 2k+3$ 이다.
  $\therefore$ $(k + 1)^2 > 2k+3$ $(= 2 (k + 1) + 1)$
  $\therefore$ $k \geq 3$인 자연수에 대해 $p(k + 1)$: $(k + 1)^2 > 2(k + 1)+1$은 참이다.

직접증명법

- 직접증명법
  - 직접증명법은 함축명제 $p \rightarrow q$가 참이 됨을 증명하는 방법
    - 주어진 유용한 정보로부터 추론을 통하여 목적하는 결론에 도달할 수 있도록 유도하는 증명 방법

직접증명법

문제
▪ 두 홀수의 곱이 홀수임을 증명하세요

- p : 두 수 m, n은 홀수다.
  q : m, n의 곱은 홀수다.
  $p \rightarrow q$: 두 홀수 m, n의 곱은 홀수다.
  두 정수 k, l 이 있을 때, 홀수 m 과 n 은 각각 $m = 2k+1$, $n = 2l+1$ 로 표현할 수 있다.
  m과 n의 곱을 표현하면 다음과 같다.
  $m \times n = (2k+1)\times(2l+1) = 4kl + 2k + 2l + 1 = 2(2kl + k + l) + 1$
  $2kl + k + l$을 변수 a로 대치하면 $m \times n = 2a + 1$로 홀수가 된다.
  ∴ 명제 $p \rightarrow q$ "두 홀수 m, n 의 곱은 홀수다"는 참이다.

직접증명법

문제

- 모든 정수 n 에 대해 n 이 짝수면, n^2 도 짝수임을 증명하세요

- p : 정수 n 은 짝수다.
  q : n^2 도 짝수다.
  $p \rightarrow q$: 정수 n 이 짝수면, n^2 도 짝수다.
  정수 k 가 있을 때, 짝수 n 은 n = 2k 로 표현할 수 있다.
  $n^2 = (2k)^2 = 4k^2 = 2(2k)^2$
  ∴ 명제 $p \rightarrow q$ "정수 n 이 짝수면, n^2 도 짝수다"는 참이다.

간접증명법

- 간접증명법
  - 함축명제 p→q를 다양한 형태로 변형하여 증명하는 방법
    - 대우증명법, 모순증명법, 반례증명법이 이에 속함

간접증명법

- 대우증명법
  - 함축명제 p→q를 다양한 형태로 변형하여 증명하는 방법
    - p → q와 ~q → ~p가 대우 관계로서 논리적 동치가 됨을 이용하여, ~q → ~p가 참인 것을 증명
    - p → q가 참이 되는 것을 논리적 동치 관계를 이용하여 간접적으로 보여주는 증명 방법

간접증명법

문제

- 모든 정수 n에 대해 n^2 이 짝수면, n 도 짝수임을 증명하세요

- p : 모든 정수 n 에 대해 n^2 은 짝수다.
  q : 정수 n 은 짝수다.
  ~ p : 모든 정수 n 에 대해 n^2 은 짝수가 아니다(홀수다).
  ~q : 정수 n 은 짝수가 아니다(홀수다).
  ~q → ~p : 정수 n 이 홀수면, n^2 은 홀수다.
  n 이 짝수가 아니므로 $n = 2k + 1$(k는 정수)이 되고, $n^2 = (2k + 1)^2 = 4k^2 + 4k +1 = 2(2k^2+2k)+1$
  이므로 n^2 역시 홀수다.
  따라서 ¬ q → ¬ p "정수 n 이 홀수면, n^2 은 홀수다"는 참이다.
  ∴ 명제 p → q "모든 정수 n 에 대해 n^2 이 짝수면, n 도 짝수다"는 참

간접증명법

- 모순증명법
  - 함축명제 $p \to q$ 가 $\sim(p \wedge \sim q)$ 와 동치라는 것을 다음과 같이 증명

    $\sim(p \wedge \sim q) \equiv \sim p \vee \sim(\sim q)$ ∵ 드모르간의 법칙

    $\equiv \sim p \vee q$ ∵ 이중부정의 법칙

    $\equiv p \to q$ ∵ 함축법칙

간접증명법

문제

- 두 홀수의 곱은 홀수가 됨을 증명하세요

- p : 두 정수 m, n 은 홀수다.
  q : m, n 의 곱은 홀수다.
  ~p : m, n 의 곱은 홀수가 아니다(짝수다).
  $p \wedge \neg p$: 두 정수 m, n 은 홀수고 m, n 의 곱은 짝수다.
  두 홀수 m 과 n 은 각각 $m=2k+1(k\in Z)$, $n=2l+1(l\ni Z)$로 정의할 수 있다.
  $mn=(2k+1)(2l+1)=4kl+2k+2l+1=2(2kl+2k+2l)+1$ 이므로 mn은 홀수이지 짝수가 아니다.
  따라서 $p \wedge \neg p$: "두 홀수 m, n의 곱은 짝수다"는 거짓이다.
  ∴ 명제 $p \rightarrow q$: "두 홀수의 곱은 홀수다"는 참이다.

간접증명법

- 반례증명법
  - 주어진 명제에 모순이 되는 예를 찾아 증명하는 방법이다.
    - $\exists x \sim p(x)$
    - 주어진 명제에서 모순이 되는 간단한 하나의 예를 보임으로써 비교적 쉽게 증명할 수 있는 방법

간접증명법

문제
▪ 모든 실수 x,y 에 대해 $x > y$ 이면 $x^2 > y^2$ 인지 증명하세요

- $x=0$, $y= -1$인 경우, $0>-1$이므로 $x > y$ 지만, $(0)^2<(-1)^2$ 이므로 $x^2 > y^2$ 은 성립하지 않는다.
  ∴ 모든 실수 x,y 에 대해 $x > y$ 이면 $x^2 > y^2$ 인 것은 아니다.

간접증명법

- 존재증명법
  - 주어진 명제가 참(T)이 되는 예를 찾아 증명하는 방법
    - $\exists x\ p(x)$ $p(x)$를 x라는 변수를 가지는 명제라고 한다면 $p(x)$가 참인 x가 적어도 하나가 존재한다는 것을 보이는 증명 방법이다.

간접증명법

문제

- 어떤 실수 x,y 에 대해 $x > y$ 이면 $x^2 > y^2$ 인지 증명하세요

- $x=2$, $y= -1$인 경우, $2>-1$이므로 $x > y$ 이 성립한다.
  $\therefore$ 어떤 실수 x,y 에 대해 $x > y$ 이면 $x^2 > y^2$ 이 성립한다.

재귀법

- 재귀법
  - 하나의 문제를 그보다 작은 값을 가지는 동일한 문제로 계속 단순화시켜 해결하는 방법

재귀법

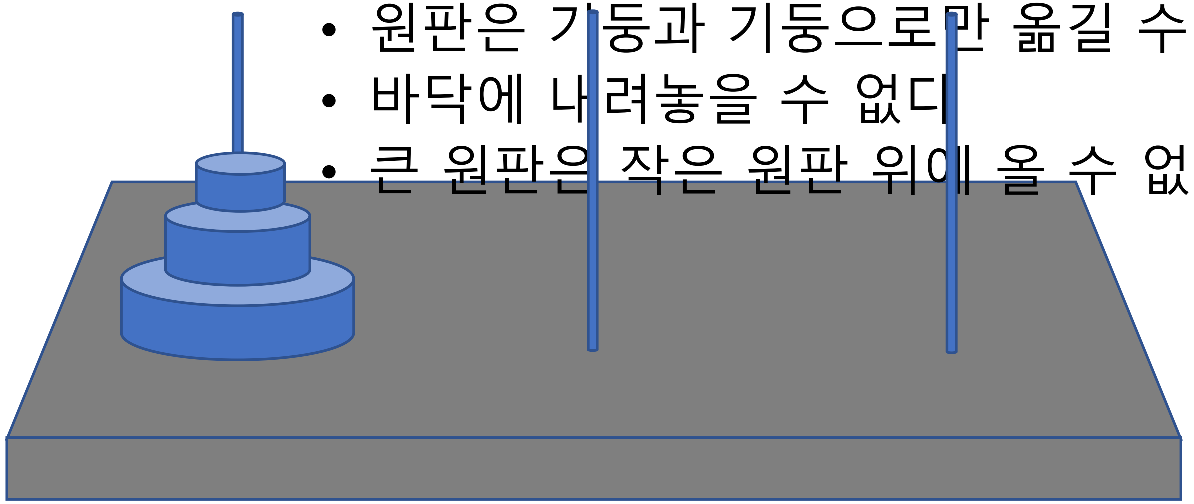
문제

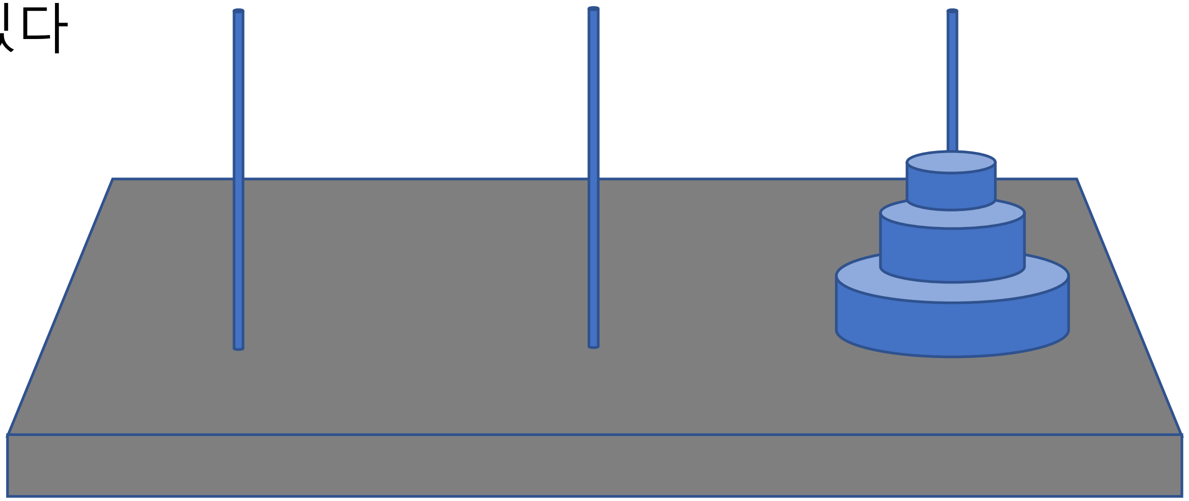
- 재귀법을 이용하여 양의 정수 5에 대한 팩토리얼 값을 구해보세요

- 5! = 5 x 4 x 3 x 2 x 1 = 5 x 4 x 3 x 2! = 5 x 4 x 3! = 5 x 4! = 5! = 120

재귀법

- 재귀법
    - 하노이의 탑
        - 바닥에 3개의 기둥이 세워져 있고 가장 왼쪽 기둥에 크기가 다른 n개의 원판이 가장 큰 원판을 맨 아래로 하여 크기가 작은 순으로 쌓여있다고 할 때 이 원판들을 가장 오른쪽 기둥으로 그대로 옮기는 과정에서의 최소 이동 횟수를 구하는 문제이다.
        - 추가조건으로 원판은 한 번에 하나씩만 옮길 수 있다
        - 원판은 기둥과 기둥으로만 옮길 수 있다
        - 바닥에 내려놓을 수 없다
        - 큰 원판은 작은 원판 위에 올 수 없다





이산수학 – 문제해결 - 파이썬코딩

- 1. A 기둥에 있는 n-1 개의 원반을 B 기둥으로 옮깁니다.
- 2. A 기둥에 남아 있는 원반중 가장 큰 원반을 C 기둥으로 옮깁니다.
- 3. B 기둥에 있는 n-1개 원반을 C 기둥으로 옮깁니다.

- 입력: 옮기려는 원반의 개수 (n)
- 옮길 원반이 현재 있는 출발점 기둥 (from_post)
- 원반을 옮기 도착점 기둥 (to_post)
- 옮기는 과정에서 사용할 보조 기둥 (aux_post)
- 출력: 원반을 옮기는 순서

- hanoi(n, from_post, to_post, aux_post):

이산수학 – 문제해결 - 파이썬코딩

```
def my_hanoi(n, from_post, to_post, aux_post):
    if n==1: #원반 한 개를 옮기다면 (옮기는 원반이 한 개 라면)
        print(from_post, "---->", to_post) # 바로 옮기면 됨
        return

    #원반 n-1개를 보조기둥으로 이동한다 (즉 to_post를 보조 기둥으로 한다)
    My_hanoi(n-1, from_post, aux_post, to_post)

    # 가장 큰 원반을 목적지로 이동한다
    print(from_post, "--→", to_post)

    #보조 기둥에 있는 원반 n-1개를 목적지로 이동한다 (즉, from_post를 보조 기둥으로 한다)
    my_hanoi(n-1, aux_post, to_post, from_post)

print("n=1인 경우")
my_hanoi(1,1,3,2)

print("n=2인 경우")
my_hanoi(2,1,3,2)

print("n=3인 경우")
my_hanoi(3,1,3,2)
```

이산수학 – 문제해결 - 파이썬코딩

```
def my_hanoi(n, from_post, to_post, aux_post):
    if n==1: #원반 한 개를 옮기다면 (옮기는 원반이 한 개 라면)
        print(from_post, "---->", to_post) # 바로 옮기면 됨
        return

    #원반 n-1개를 보조기둥으로 이동한다 (즉 to_post를 보조 기둥으로 한다)
    my_hanoi(n-1, from_post, aux_post, to_post)

    #가장 큰 원반을 목적지로 이동한다
    print(from_post,"---->",to_post)

    #보조 기둥에 있는 원반 n-1개를 목적지로 이동한다(즉, from_post를 보조 기둥으로 한다)
    my_hanoi(n-1,aux_post, to_post, from_post)

def hanoi_test():
    num = input("원반의 개수를 입력하세요")
    print("원반의 개수",num)
    my_hanoi(int(num),1,3,2)

hanoi_test()
```

이산수학 – 문제해결 - 파이썬코딩

```
def my_factorial(n):
    if n == 0:
        return 1
    return n * my_factorial(n-1)

def factorial_test():
    num = input("팩토리얼 수를 입력하세요")
    print("팩토리얼의 수",num)
    print(my_factorial(int(num)))

factorial_test()
```

이산수학 – 문제해결 - 파이썬코딩

```
def sum_test():
    num = input("수를 입력하세요")
    print("입력한 수",num)
    print(my_sum(int(num)))

def my_sum(n): #  0~n 까지의 합을 구하는 것
    if n == 0: # n=0 이면 합은 0
        return 0
    return n + my_sum(n-1) # n이 0보다 크면 0에서 n 까지의 합은, n-1 까지의 합에 n을 더한 것

sum_test()
```

이산수학 – 문제해결 - 파이썬코딩

```
import turtle as t

def my_spiral(sp_len):
    if sp_len <= 5:
        return
    t.forward(sp_len)
    t.right(90)
    my_spiral(sp_len - 5)

t.speed(0)
my_spiral(200)
t.hideturtle()
t.done()
```

학습 내용 요약

- 증명
  - 수학적귀납법
  - 직접증명
  - 간접증명
  - 재귀법