4471028: 프로그래밍언어

Lecture 3 — 귀납 증명
Inductive Proof

임현승

2020 봄학기

수학적 귀납법(Mathematical Induction)

자연수에 대한 성질을 증명하기 위한 기법

$n \geq b$를 만족하는 모든 자연수에 대해 명제(proposition) $P(n)$이 성립함을 보이려면 다음을 증명하면 된다.

1. (Base case) $P(b)$가 성립함을 보인다.
2. (Inductive case) $P(n)$이 성립함을 가정하고 $P(n+1)$이 성립함을 보인다. 이때, 가정 $P(n)$은 *귀납 가정(induction hypothesis)*이라고 불린다.

예제 1

Prove that $P(n)$ is true for every $n \in \mathbb{N}$ such that $n \geq 1$.

$$P(n) \quad \triangleq \quad \Sigma_{k=1}^{n} k = \frac{n(n+1)}{2}$$

Proof. By mathematical induction on n

- (Base case) $n = 1$
  - $\Sigma_{k=1}^{1} k = 1 = \frac{1(1+1)}{2}$ by simple calculation
- (Inductive case) $n = m + 1$
  - 증명해야 되는 명제: If $P(m)$ holds, then $P(m+1)$ also holds.
  - 귀납 가정(IH) $P(m)$: $\Sigma_{k=1}^{m} k = \frac{m(m+1)}{2}$
  - We prove $P(m+1)$: $\Sigma_{k=1}^{m+1} k = \frac{(m+1)(m+2)}{2}$ as follows:

$$\begin{aligned}
\Sigma_{k=1}^{m+1} k &= \Sigma_{k=1}^{m} k + (m+1) \\
&= \frac{m(m+1)}{2} + (m+1) && \text{by IH} \\
&= \frac{(m+1)(m+2)}{2}
\end{aligned}$$

예제 2

Prove that $P(n)$ is true for every $n \in \mathbb{N}$ s.t. $n \geq 4$.

$$P(n) \quad \triangleq \quad n! > 2^n$$

Proof. By mathematical induction on n

- (Base case) $n = 4$
  - $4! = 24 > 16 = 2^4$ by simple calculation
- (Inductive case) $n = k + 1$
  - If $P(k)$ holds, then $P(k+1)$ also holds $(k \geq 4)$.
  - IH $P(k)$: $k! > 2^k$
  - We prove $P(k+1)$: $(k+1)! > 2^{k+1}$ as follows:

$$\begin{aligned}
(k+1)! &= (k+1) \times k! & \\
&> (k+1) \times 2^k & \text{by IH} \\
&> 2 \times 2^k = 2^{k+1} & \text{from } k+1 > 2
\end{aligned}$$

□

구조 귀납법(Structural Induction)

귀납적으로 정의된 집합에 대한 성질을 증명하기 위한 기법

귀납적으로 정의된 집합 S의 모든 원소 s에 대해 명제 $P(s)$가 성립함을 보이려면 다음을 증명하면 된다.

1. (Base case) 하위 구조(substructure)가 없는, 즉 공리(axiom)로 정의되는 모든 원소 b에 대해 $P(b)$가 성립함을 보인다.
2. (Inductive case) 귀납적으로 정의되는 구조 s에 대해, s의 모든 하위 구조 s'에 대해 $P(s')$임이 성립함을 가정하고(귀납 가정), 이를 이용하여 $P(s)$가 성립함을 보인다.

예제 1

집합 S의 원소는 다음과 같이 귀납적으로 정의된다.

$$x, y \ ::= \ 3 \mid x + y$$

S의 모든 원소는 3으로 나누어질 수 있음을 증명하라.

Proof. By structural induction.

- (Base case) $x = 3$. Obviously, x is divisible by 3.
- (Inductive case) $x = y_1 + y_2$
  The induction hypotheses are

  $$y_1 \text{ is divisible by } 3, \qquad y_2 \text{ is divisible by } 3.$$

  Let $y_1 = 3k_1$ and $y_2 = 3k_2$. Using IHs, we derive

  $$y_1 + y_2 \text{ is divisible by } 3$$

  as follows:

  $$\begin{array}{rcll} y_1 + y_2 & = & 3k_1 + 3k_2 & \text{by IHs} \\ & = & 3(k_1 + k_2) & \end{array}$$

□

예제 2

다음 추론 규칙에 의해 정의되는 집합 S를 고려하자.

$$\frac{}{() \in S} \qquad \frac{x \in S}{(x) \in S} \qquad \frac{x \in S \quad y \in S}{xy \in S}$$

S의 모든 원소는 같은 수의 '('와 ')'로 구성됨을 보여라.

Proof 증명하고자 하는 명제를 엄밀하게 작성하면:

$$\text{If } x \in S \text{ then } l[x] = r[x]$$

$l[x]$와 $r[x]$는 각각 x에 쓰이는 '('와 ')'의 개수를 계산하는 함수; 다음과 같이 재귀적으로(recursively) 정의됨:

$$\begin{array}{rcl} l[()] & = & 1 \\ l[(x)] & = & 1 + l[x] \\ l[xy] & = & l[x] + l[y] \end{array} \qquad \begin{array}{rcl} r[()] & = & 1 \\ r[(x)] & = & 1 + r[x] \\ r[xy] & = & r[x] + r[y] \end{array}$$

예제 2

증명은 구조 귀납을 이용해서

- (Base case) $x = ()$. $l[x] = 1 = r[x]$이므로 증명 끝.
- (Inductive case) 귀납적으로 정의되는 경우는 두 가지:

$$\frac{x \in S}{(x) \in S} \qquad \frac{x \in S \quad y \in S}{xy \in S}$$

귀납 가정: $l[x] = r[x]$, $l[y] = r[y]$

Case 1) We prove $l[(x)] = r[(x)]$:

$$\begin{array}{rcll} l[(x)] & = & l[x] + 1 & \text{by definition of } l[(x)] \\ & = & r[x] + 1 & \text{by IH} \\ & = & r[(x)] & \text{by definition of } r[(x)] \end{array}$$

Case 2) We prove $l[xy] = r[xy]$:

$$\begin{array}{rcll} l[xy] & = & l[x] + l[y] & \text{by definition of } l[xy] \\ & = & r[x] + r[y] & \text{by IH} \\ & = & r[xy] & \text{by definition of } r[xy] \end{array}$$

□

예제 3

다음과 같이 정의되는 이진 나무들의 집합 T를 고려하자.

$$\frac{}{\texttt{leaf} \in T} \qquad \frac{t_1 \in T \quad t_2 \in T}{(n, t_1, t_2) \in T} \; (n \in \mathbb{Z})$$

T의 모든 원소에 대해 리프 노드(leaf node)의 수가 내부 노드(internal node)의 수보다 하나 더 많음을 보여라.

Proof. 증명하고자 하는 명제를 엄밀하게 작성하면:

$$\text{If } t \in T \text{ then } l(t) = i(t) + 1$$

$l(t)$와 $i(t)$는 각각 t의 리프 노드의 개수와 내부 노드의 개수를 계산하는 함수; 다음과 같이 재귀적으로 정의됨:

$$\begin{array}{rlrl} l(\texttt{leaf}) &= 1 & i(\texttt{leaf}) &= 0 \\ l((n, t_1, t_2)) &= l(t_1) + l(t_2) & i((n, t_1, t_2)) &= 1 + i(t_1) + i(t_2) \end{array}$$

예제 3

증명은 구조 귀납을 이용해서

- (Base case) $t = \texttt{leaf}$. $l(t) = 1$이고 $i(t) = 0$이므로 증명 끝.
- (Inductive case) $t = (n, t_1, t_2)$
  귀납 가정:
  $$l(t_1) = i(t_1) + 1, \quad l(t_2) = i(t_2) + 1.$$
  귀납 가정을 이용하여 $l((n, t_1, t_2)) = i((n, t_1, t_2)) + 1$을 다음과 같이 증명:
  $$\begin{array}{rcll} l((n, t_1, t_2)) & = & l(t_1) + l(t_2) & \\ & = & i(t_1) + 1 + i(t_2) + 1 & \text{by IH} \\ & = & i((n, t_1, t_2)) + 1 & \end{array}$$

요약

- 구조적 귀납법(Structural Induction)
  - 귀납법을 이용하여 정의된 집합에 대한 여러 가지 성질을 증명하는데 사용하는 기법
  - 집합의 원소의 구조(생김새)를 분석
  - cf. 규칙 귀납법(rule induction)은 추론 규칙에 귀납법을 적용