SNU 4190.310 Programming Languages
(slides 2-1)

Prof. Kwangkeun Yi

귀납법 = 집합의 정의

집합을 정의하는 방법

- 되돌아서 return 歸, 바치다 dedicate 納.
- 그 집합의 원소를 가지고 그 집합의 원소를 만든다.

그 집합은 이것이다

- 규칙: 가정들 X와 결론 x.
- "X에 있는 것들이 정의하려는 집합에 모두 있으면, x도 있어야 한다."
- 그러한 집합 중에서 가장 작은 집합.
  - 규칙 (X, x)들의 모임을 Φ라고 하자.
  - "Φ에 대해서 닫혀있는"(Φ-closed), "Φ-닫힌" 집합 A:

$$(X, x) \in \Phi \text{ 이고 } X \subseteq A \text{ 이면 } x \in A$$

  - Φ가 정의하는 집합: 모든 Φ-닫힌 집합들의 교집합

$$\bigcap \{A \mid A\text{는 } \Phi\text{-닫힘}\}.$$

자연수의 집합:

$$(\emptyset, 0) \quad (\{n\}, n+1)$$

영문 소문자 알파벳으로 만들어 지는 스트링의 집합:

$$(\emptyset, \epsilon) \quad (\{\alpha\}, x\alpha \text{ for } x \in \{\mathsf{a}, \cdots, \mathsf{z}\})$$

귀납규칙 표기법

자연수 집합은

$$n \;\to\; 0 \mid n+1$$

혹은

$$\frac{}{0} \qquad \frac{n}{n+1}$$

위의 스트링 집합은:

$$\begin{array}{rcll} \alpha & \to & \epsilon & \\ & \mid & x\alpha & (x \in \{\texttt{a}, \cdots, \texttt{z}\}) \end{array}$$

혹은

$$\frac{}{\epsilon} \qquad \frac{\alpha}{x\alpha}\; x \in \{\texttt{a}, \cdots, \texttt{z}\}$$

집합이 유한하다면 재귀가 필요없다. 집합 $\{1, 2, 3\}$을 규칙들로 표현하면

$$(\emptyset, 1) \quad (\emptyset, 2) \quad (\emptyset, 3)$$

혹은

$$x \to 1 \mid 2 \mid 3$$

혹은

$$\frac{}{1} \quad \frac{}{2} \quad \frac{}{3}$$

가 된다.

리스트의 집합:

$$\frac{}{\texttt{nil}} \qquad \frac{\ell}{\circ\!-\!\ell}$$

혹은

$$\ell \;\rightarrow\; \texttt{nil} \mid \circ\!-\!\ell$$

말단에 정수를 가지는 두갈래 나무(*binary tree*)들의 집합:

$$\frac{}{n}\ n \in \mathbb{Z} \qquad \frac{t}{\mathtt{N}(t, \mathtt{nil})}$$

$$\frac{t}{\mathtt{N}(\mathtt{nil}, t)} \qquad \frac{t_1 \quad t_2}{\mathtt{N}(t_1, t_2)}$$

혹은

$$\begin{array}{rcll} t & \to & n & (n \in \mathbb{Z}) \\ & \mid & \mathtt{N}(t, \mathtt{nil}) & \\ & \mid & \mathtt{N}(\mathtt{nil}, t) & \\ & \mid & \mathtt{N}(t, t) & \end{array}$$

그래프(*graph*)는 노드들과 노드를 연결하는 선들로 이루어진 구조물이다. 그래프들의 집합:

$$\frac{}{\circ} \qquad \frac{G}{G\circ} \qquad \frac{G}{G\supset}$$

혹은

$$\begin{array}{rcll} G & \to & \circ & \text{a node} \\ & | & G\circ & \text{adding a node} \\ & | & G\supset & \text{adding an edge} \end{array}$$

"$G\supset$"은 그래프 G에 있는 노드를 선(*edge*)으로 연결한 것.

정수식들의 집합:

$$\frac{}{n}\; n \in \mathbb{N} \qquad \frac{e}{-e}$$

$$\frac{e_1 \quad e_2}{e_1 + e_2} \qquad \frac{e_1 \quad e_2}{e_1 * e_2}$$

혹은

$$\begin{array}{rcll} e & \rightarrow & n & (n \in \mathbb{N}) \\ & \mid & -e & \\ & \mid & e + e & \\ & \mid & e * e & \end{array}$$

그 집합은 이렇게 만든다

규칙 집합 Φ는 함수 ϕ를 정의:

$$\phi(Y) = \{x \mid \frac{X}{x} \in \Phi,\ X \subseteq Y\}$$

Φ 규칙들이 정의하는 집합은 함수 ϕ에 의해서 닫혀있는 집합중에서

$$\bigcap\{X|\phi(X) \subseteq X\}$$

이다. (이 집합은 ϕ의 최소고정점(least fixed point)이다.)

귀납 규칙이 정의하는 집합

$$\bigcap\{X|\phi(X) \subseteq X\}$$

을 이렇게 만들 수 있다:

$$\phi^0 \cup \phi^1 \cup \phi^2 \cup \cdots = \bigcup_{i \in \mathbb{N}} \phi^i$$

여기서

$$\begin{array}{rcl} \phi^0 & = & \emptyset \\ \phi^n & = & \phi(\phi^{n-1}) \quad n \in \mathbb{N} \end{array}$$

자연수 집합 규칙

$$n \;\rightarrow\; 0 \mid n+1$$

그 집합은:

$$\begin{aligned} \phi^0 &= \emptyset \\ \phi^1 &= \{0\} \\ \phi^2 &= \{0,1\} \\ \phi^3 &= \{0,1,2\} \\ \ldots \end{aligned}$$

들의 합집합.

리스트의 집합 규칙

$$\ell \ \rightarrow \ \texttt{nil} \mid \circ\!-\!\ell$$

그 집합은:

$$\begin{array}{lcl}
\phi^0 & = & \emptyset \\
\phi^1 & = & \{\texttt{nil}\} \\
\phi^2 & = & \{\texttt{nil}, \circ-\texttt{nil}\} \\
\phi^3 & = & \{\texttt{nil}, \circ-\texttt{nil}, \circ-\circ-\texttt{nil}\} \\
\ldots & &
\end{array}$$

들의 합집합.

두갈래 나무(*binary tree*)의 집합 규칙:

$$\begin{array}{rcl} t & \to & \circ \\ & | & \mathtt{N}(t, \mathtt{nil}) \\ & | & \mathtt{N}(\mathtt{nil}, t) \\ & | & \mathtt{N}(t, t) \end{array}$$

이 집합은:

$$\begin{array}{rcl} \phi^0 & = & \emptyset \\ \phi^1 & = & \{\circ\} \\ \phi^2 & = & \{\circ, \mathtt{N}(\circ, \mathtt{nil}), \mathtt{N}(\mathtt{nil}, \circ), \mathtt{N}(\circ, \circ)\} \\ \ldots & & \end{array}$$

들의 합집합.

스트링 집합 규칙:

$$\begin{array}{rcl} \alpha & \to & \epsilon \\ & \mid & x\alpha \quad (x \in \{\mathtt{a}, \cdots, \mathtt{z}\}) \end{array}$$

이 집합은

$$\begin{array}{rcl} \phi^0 & = & \emptyset \\ \phi^1 & = & \{\epsilon\} \\ \phi^2 & = & \{\epsilon, \mathtt{a}\epsilon, \cdots, \mathtt{z}\epsilon\} \\ \ldots & & \end{array}$$

들의 합집합.

그래프 집합:

$$
\begin{array}{rcll}
G & \rightarrow & \circ & \textsf{a node} \\
& | & G\circ & \textsf{adding a node} \\
& | & G\supset & \textsf{adding an edge}
\end{array}
$$

이 집합은

$$
\begin{array}{rcl}
\phi^0 & = & \emptyset \\
\phi^1 & = & \{\circ\} \\
\phi^2 & = & \{\circ, \circ\circ, \circ\supset\} \\
\ldots & &
\end{array}
$$

들의 합집합.

밑바닥 없는 규칙

집합 규칙:

$$
\begin{array}{rcl}
t & \rightarrow & \mathtt{N}(t, \mathtt{nil}) \\
 & | & \mathtt{N}(\mathtt{nil}, t) \\
 & | & \mathtt{N}(t, t)
\end{array}
$$

이 집합은 $\emptyset$:

$$\phi^0 = \phi^1 = \cdots = \emptyset$$

모든 귀납 규칙

$$\frac{X}{x}$$

에서 $X \neq \emptyset$이면 정의하는 집합은 $\emptyset$

정리

- 귀납법 = 집합의 정의 inductive definition
- 표현 방법들
- 그 집합은 무엇이지?
  - "닫혀있는 최소의 집합"
  - $\bigcap\{A | A\text{는 } \Phi\text{-닫힘}\}$
- 그 집합은 어떻게 만들지?
  - "$\emptyset$에서 출발해서, 규칙이 만드는 원소들만을 빠짐없이 첨가"
  - $\bigcup_{i\in\mathbb{N}} \phi^i$

계획

- 귀납법 = 증명 proof by induction
  - 원소의 순서
  - "모든 원소들" = "모든 순서의 원소들"
- 형식논리와 추론
  - 논리식 집합의 정의 (귀납법)
  - 논리식 의미의 정의 (조립식)
  - 참논리식 집합의 정의 (귀납법)
  - 참논리식 추론의 방법 (귀납법)
  - 추론방법 평가하기

원소들의 순서

귀납적으로 정의된 집합 S

- ϕ^i번째에 새롭게 만들어 지는 원소: i째번 원소.
- 0째번에 만들어진 원소들이 씨.
- "기초가 튼튼한 순서(*well-founded order*)"
- 귀납적으로 정의된 집합은 기초가 튼튼한 순서를 가지고 있다.

귀납법 = 증명의 방법

증명 목표:

$$\forall x \in S.P(x)$$

- S가 귀납적으로 정의됨, 즉, 모든 원소들의 순서가 있슴.
- P(0째번 원소)를 증명: 항상 성립
- 임의의 $i > 0$에 대해서

$$(\forall j < i.P(j\text{째번 원소})) \Rightarrow P(i\text{째번 원소})$$

  를 증명.

("대우법 증명"도 있슴)

귀납 증명: 규칙들에 대한 것으로

임의의 $i > 0$에 대해서

$$(\forall j < i. P(j\text{째번 원소})) \Rightarrow P(i\text{째번 원소})$$

를 증명.

- case $i = 1$: $P(1\text{째번 원소})$ 증명
- case $i > 1$: $(\forall j < i. P(j\text{째번 원소})) \Rightarrow P(i\text{째번 원소})$ 증명

즉,

- "base case": 규칙 $\frac{}{x}$가 만드는 x들에 대해, $P(x)$ 증명.
- "inductive case": 규칙 $\frac{X}{x}$가 만드는 x들에 대해, $(\forall a \in X. P(a)) \Rightarrow P(x)$를 증명.

귀납증명 예

- 증명: $\forall n \in \mathbb{N}. 0 + 1 + 2 + \cdots + n = n(n+1)/2$
- 증명: 모든 두갈래가 꽉찬 나무(*complete binary tree*)는 말단 노드의 갯수는 내부 노드의 갯수보다 하나 많다.

$$t \to \circ \mid \mathtt{N}(t, t)$$

귀납증명 예

다음 두 가지 방법으로 정의되는 집합(그래프 집합)은 같은 집합이다.

$$
\begin{array}{rcl}
G & \rightarrow & \circ \mid G \circ \mid G\supset \\
H & \rightarrow & \circ \mid H\ H \mid H\supset
\end{array}
$$

우선, 임의의 G는 H로 볼 수 있다. G를 만드는 세가지 규칙마다 귀납적으로 쉽게 확인해볼 수 있다.
거꾸로, 임의의 H도 G로 볼 수 있다. H에 대한 귀납법으로 증명. (귀납법이 두 겹으로 필요.)

논리식 집합

귀납적 정의

$$
\begin{array}{rcl}
f & \rightarrow & T \mid F \\
 & \mid & \neg f \\
 & \mid & f \wedge f \\
 & \mid & f \vee f \\
 & \mid & f \Rightarrow f
\end{array}
$$

논리식 의미

조립식 정의 compositional definition

$$
\begin{array}{rcl}
[\![T]\!] & = & \textsf{true} \\
[\![F]\!] & = & \textsf{false} \\
[\![\neg f]\!] & = & \textsf{not}[\![f]\!] \\
[\![f_1 \wedge f_2]\!] & = & [\![f_1]\!] \ \textsf{andalso} \ [\![f_2]\!] \\
[\![f_1 \vee f_2]\!] & = & [\![f_1]\!] \ \textsf{orelse} \ [\![f_2]\!] \\
[\![f_1 \Rightarrow f_2]\!] & = & [\![f_1]\!] \ \textsf{implies} \ [\![f_2]\!]
\end{array}
$$

임의의 논리식 f의 의미가 정의 된 셈.

$$
\begin{aligned}
& [\![(T \wedge (T \vee F)) \Rightarrow F]\!] \\
= \quad & [\![T \wedge (T \vee F)]\!] \ \textsf{implies} \ [\![F]\!] \\
= \quad & ([\![T]\!] \ \textsf{andalso} \ [\![T \vee F]\!]) \ \textsf{implies false} \\
= \quad & (\textsf{true andalso} \ ([\![T]\!] \ \textsf{orelse} \ [\![F]\!])) \ \textsf{implies false} \\
= \quad & (\textsf{true andalso} \ (\textsf{true orelse false})) \ \textsf{implies false} \\
= \quad & \textsf{false}
\end{aligned}
$$

집합의 정의, 추론 규칙

- 쌍 uncle(a, b)들의 집합

$$\frac{\textsf{male}(u) \quad \textsf{father}(f, i) \quad \textsf{brother}(f, u)}{\textsf{uncle}(u, i)}$$

$$\frac{\textsf{father}(c, a) \quad \textsf{father}(c, b)}{\textsf{brother}(a, b)}$$

$$\cdots$$

- 쌍 append(a, b, c)들의 집합

$$\frac{}{\textsf{append}(l, [], l)} \qquad \frac{}{\textsf{append}([], l, l)}$$

$$\frac{\textsf{append}(r, b, c)}{\textsf{append}(a :: r, b, a :: c)}$$

추론 과정 = 계산 (또는 "증명 나무")

$$\cfrac{\cfrac{\cfrac{}{\mathsf{append}([], [3,4], [3,4])}}{\mathsf{append}([2], [3,4], [2,3,4])}}{\mathsf{append}([1,2], [3,4], [1,2,3,4])}$$

어떤 집합의 정의, 추론 규칙

쌍 $(\{g_1, \cdots, g_n\}, f)$들의 집합

$$\frac{}{(\Gamma, T)} \qquad \frac{}{(\Gamma, f)}\ f \in \Gamma \qquad\qquad \frac{(\Gamma, F)}{(\Gamma, f)} \qquad \frac{(\Gamma, \neg\neg f)}{(\Gamma, f)}$$

$$\frac{(\Gamma, f_1) \quad (\Gamma, f_2)}{(\Gamma, f_1 \wedge f_2)} \qquad\qquad \frac{(\Gamma, f_1 \wedge f_2)}{(\Gamma, f_1)}$$

$$\frac{(\Gamma, f_1)}{(\Gamma, f_1 \vee f_2)} \qquad\qquad \frac{\begin{array}{c}(\Gamma, f_1 \vee f_2) \\ (\Gamma \cup \{f_1\}, f_3) \quad (\Gamma \cup \{f_2\}, f_3)\end{array}}{(\Gamma, f_3)}$$

$$\frac{(\Gamma \cup \{f_1\}, f_2)}{(\Gamma, f_1 \Rightarrow f_2)} \qquad\qquad \frac{(\Gamma, f_1 \Rightarrow f_2) \quad (\Gamma, f_1)}{(\Gamma, f_2)}$$

$$\frac{(\Gamma \cup \{f\}, F)}{(\Gamma, \neg f)} \qquad\qquad \frac{(\Gamma, f) \quad (\Gamma, \neg f)}{(\Gamma, F)}$$

형식논리의 표기법으로

$$\frac{}{\Gamma \vdash T} \qquad \frac{}{\Gamma \vdash f}\ f \in \Gamma \qquad \frac{\Gamma \vdash F}{\Gamma \vdash f} \qquad \frac{\Gamma \vdash \neg\neg f}{\Gamma \vdash f}$$

$$\frac{\Gamma \vdash f_1 \quad \Gamma \vdash f_2}{\Gamma \vdash f_1 \wedge f_2} \qquad \frac{\Gamma \vdash f_1 \wedge f_2}{\Gamma \vdash f_1}$$

$$\frac{\Gamma \vdash f_1}{\Gamma \vdash f_1 \vee f_2} \qquad \frac{\begin{array}{c}\Gamma \vdash f_1 \vee f_2 \\ \Gamma \cup \{f_1\} \vdash f_3 \quad \Gamma \cup \{f_2\} \vdash f_3\end{array}}{\Gamma \vdash f_3}$$

$$\frac{\Gamma \cup \{f_1\} \vdash f_2}{\Gamma \vdash f_1 \Rightarrow f_2} \qquad \frac{\Gamma \vdash f_1 \Rightarrow f_2 \quad \Gamma \vdash f_1}{\Gamma \vdash f_2}$$

$$\frac{\Gamma \cup \{f\} \vdash F}{\Gamma \vdash \neg f} \qquad \frac{\Gamma \vdash f \quad \Gamma \vdash \neg f}{\Gamma \vdash F}$$

증명 나무

$$
\dfrac{
  \dfrac{}{\{p \to \neg p, p\} \vdash p}
  \qquad
  \dfrac{
    \dfrac{}{\{p \to \neg p, p\} \vdash p \to \neg p}
    \qquad
    \dfrac{}{\{p \to \neg p, p\} \vdash p}
  }{\{p \to \neg p, p\} \vdash \neg p}
}{
  \dfrac{\{p \to \neg p, p\} \vdash F}{\{p \to \neg p\} \vdash \neg p}
}
$$

또 다른 시선: 증명(증명나무)들의 집합을 정의

증명들의 집합을 만드는 귀납규칙 (“증명규칙” inference rules).

- 예를 들어, 증명규칙

$$\frac{\Gamma \vdash f_1 \quad \Gamma \vdash f_2}{\Gamma \vdash f_1 \wedge f_2}$$

  은 증명을 만드는 귀납 규칙
- $\Gamma \vdash f_1$와 $\Gamma \vdash f_2$의 증명들을 가지고 $\Gamma \vdash f_1 \wedge f_2$의 증명을 만든다.

증명 규칙의 평가

기계가 만드는 $\{g_1, \cdots, g_n\} \vdash f$ 는 어떤 것들인가? 예)
$[\![g_1 \wedge \cdots \wedge g_n \Rightarrow f]\!] = \mathsf{true}$ 인가?

- 기계의 안전성 soundness:

$$\Gamma \vdash f \text{ 이면 } [\![\Gamma \Rightarrow f]\!] = \mathsf{true}$$

- 기계의 완전성 completeness:

$$\Gamma \vdash f \text{ 면이 } [\![\Gamma \Rightarrow f]\!] = \mathsf{true}$$