SNU 프로그래밍언어 특강

(1.0)

이 광근

kwangkeunyi.snu.ac.kr

계획

무엇 의미구조$_{\text{denotational semantics}}$ 세계에서 증명방법

- 고정점에 대한 증명법
  - 고정점 인덕$_{\text{fixpoint induction}}$으로 증명하기
  - 고정점 정의를 이용해서 증명하기
- 겉보기 증명$_{\text{extentional proof}}$ 스타일

(미리보기) 실행 의미구조$_{\text{operational semantics}}$ 세계에서의 증명 대비 장점

무엇 의미구조$_{\text{denotational semantics}}$ 증명법 I: 고정점에 대한

- 프로그램 의미는 연속함수의 최소고정점
- 프로그램 의미의 성질 = 최소고정점의 성질
- 최소고정점의 성질 증명 방법
  1. 고정점 인덕$_{\text{fixpoint induction}}$으로 (만능 아님)
  2. 고정점 정의를 이용하기

고정점 인덕$_{\text{fixpoint induction}}$

- CPO D, 연속함수 $f : D \to D$

$$\mathit{lfp}\, f = \bigsqcup C \quad (C \stackrel{\text{let}}{=} \{f^i \bot_D \mid i \in \mathbb{N}\})$$

증명할것: $P(\mathit{lfp}\, f)$ (성질 P)

- $\forall x \in C.P(x)$를 보이면, $P(\mathit{lfp}\, f)$을 보인것임?
- 아님; $\mathit{lfp}\, f \notin C$ 인 경우가 흔하다(무한 CPO)

단, 성질 P가 품에넣는 성질$_{\text{inclusive assertion}}$인 경우라면 오케이:

- 고정점 체인 C 집합을 만드는 인덕규칙:

$$\frac{}{\perp} \qquad \frac{x}{f(x)}$$

  이므로

- $P(\perp)$임을 보이고
- $P(x) \Rightarrow P(f(x))$을 보이면,
- $P(\mathit{lfp}\, f)$도 사실임 ("$\mathit{lfp}\, f$도 그 품에 들어감").

품에넣는 성질 P: 관심 체인 $C \subseteq D$에 대해서

$$(\forall x \in C. P(x)) \Longrightarrow P(\bigsqcup C)$$

인 성질.

- P가 품에넣는 성질? 생김새로 판단하기

$$
\begin{array}{rcl}
P & \to & P \land P \mid \forall \vec{y}.EP \\
EP & \to & EP \lor EP \mid Q(\vec{y}) \mid f(\vec{y}) \sqsubseteq g(\vec{y}) \\
Q & & \text{1차 논리식}_{\text{first order predicate}} \\
f, g & & \text{연속함수}
\end{array}
$$

(p.215, *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory*, Joseph E. Stoy)

("Inductive Method for Proving Properties of Programs", Manna, Ness, Vuillemin)

- P가 품에넣는 성질? 내용으로 판단하기
  - 관심 체인 $C \subseteq D$에 대해서

$$(\forall x \in C.P(x)) \implies P(\bigsqcup C)$$

    인지 확인
  - 예) 연속함수 $f, g \in D \to D$, 증명할것: $P(\mathit{lfp}\, f, \mathit{lfp}\, g)$,

$$P(a, b) \stackrel{\text{let}}{=} \forall x.(a(x) \sqsubseteq k_1 \implies b(x) \sqsubseteq k_2).$$

    위의 P는 품에넣는 성질이다
    - 관심 체인 $C = \{(f^i \bot, g^i \bot) \mid i \in \mathbb{N}\}$에 대해

$$(\forall (a, b) \in C.P(a, b)) \implies P(\bigsqcup C)$$

      이므로. (왜?)

고정점 인덕$_{\text{fixpoint induction}}$ 예

$$
\begin{aligned}
&[\![\texttt{while}\, E\, C]\!] \\
&\quad = \quad \mathit{lfp}\lambda X.(\lambda M.[\![E]\!]M ?\, X([\![C]\!]M) \; : \; M) \\
&[\![\texttt{repeat} \;\; C \;\; E]\!] \\
&\quad = \quad \mathit{lfp}\lambda X.((\lambda M.[\![E]\!]M ?\, X M \; : \; M) \circ [\![C]\!])
\end{aligned}
$$

증명: $[\![C \; ; \; \texttt{while}\; E\; C]\!] = [\![\texttt{repeat} \;\; C \;\; E]\!]$ 즉,

$$
\begin{array}{rcl}
[\![C \,;\, \texttt{while}\, E\, C]\!] & = & (\mathit{lfp} F) \circ [\![C]\!] \\
[\![\texttt{repeat}\ \ C\ \ E]\!] & = & \mathit{lfp} G \\
F, G & = & \cdots
\end{array}
$$

이므로, 증명할 것은

$$
P(\mathit{lfp} F, \mathit{lfp} G) \stackrel{\text{let}}{=} (\mathit{lfp} F \circ [\![C]\!] = \mathit{lfp} G).
$$

- 품에넣는 성질$_{\text{inclusive predicate}}$ 이므로
- 고정점 인덕$_{\text{fixpoint induction}}$으로:

  $P(\bot, \bot)$를 보이고, $P(f, g) \Rightarrow P(F(f), G(g))$를 보인다.

고정점 정의를 이용한 증명법

최소고정점 정의를 이용해서

$$lfp\, f \stackrel{\text{def}}{=} \bigsqcap\{x \mid f(x) \sqsubseteq x\}$$

- 조건 $f(A) \sqsubseteq A$를 만족하는 A를 찾고
- $P(A)$ 임을 보이면, $P(lfp\, f)$이 사실이다.
  - 단, P가 $(P(x) \ \wedge \ y \sqsubseteq x \implies P(y))$이어야
- 왜: $lfp\, f$은 그런 A보다 작은 원소이므로.

고정점 정의를 이용하는 증명 예

```
f(n) = if n=0 then ∞ else (f(n-1) || true)
```

- 증명할 것: $[\![\texttt{f}]\!] \sqsubseteq \lambda x.true$
- 즉, $lfp(F \stackrel{\text{let}}{=} \lambda f.\lambda n.n = 0?\bot : (f(n-1)\ or^*\ true))$
  $\sqsubseteq \lambda x.true$.

증명하기:

1. 찾아라 $F(f) \sqsubseteq f$ 인 f, 그리고
2. f가, 확인하려는 성질을 만족함을 확인하라: $f \sqsubseteq \lambda x.true$, 그러면
3. $lfp F \sqsubseteq \lambda x.true$이다.

그런 f는 $(\lambda n.n = 0?\bot : true)$

고정점 정의를 이용하는 증명 예

```
f(n) = if n=0 then ∞ else (f(n-1) || true)
```

- 증명할 것: $[\![\texttt{f}]\!] \sqsubseteq \lambda x.true$
- 즉, $lfp(F \stackrel{\text{let}}{=} \lambda f.\lambda n.n = 0?\bot : (f(n-1)\ or^*\ true))$
  $\sqsubseteq \lambda x.true.$

증명하기:

1. 찾아라 $F(f) \sqsubseteq f$ 인 f, 그리고
2. f가, 확인하려는 성질을 만족함을 확인하라:
   $f \sqsubseteq \lambda x.true$, 그러면
3. $lfp F \sqsubseteq \lambda x.true$이다.

그런 f는 $(\lambda n.n = 0?\bot : true)$

(참고) 실행 의미구조$_{\text{operational semantics}}$ 세계에서 증명한다면 상대적으로 지루함. 인덕 증명 + 실행과정 추적: $\forall$n $\in \mathbb{N}$.f(n)이 안멈추거나 true 값을 계산함.

무엇 의미구조$_{\text{denotational semantics}}$ 증명법 II: 겉보기 증명$_{\text{extentional proof}}$ 스타일

확인하고자 하는 성질 P가 의미 세계안에서 드러나는 "겉모습"에 관한 것일때.

- 예) 프로그램변환 맞음 $\Rightarrow P([\![\text{출발프로그램}]\!], [\![\text{도착프로그램}]\!])$.
  - $P(a,b) \stackrel{\text{def}}{=} a = b$ (겉모습 $=$ 무엇의미가 같음)
  - $P(a,b) \stackrel{\text{def}}{=} a \sim b$ (겉모습 $=$ 무엇의미 세계에서 어떤 관계)
- 겉모습 증명의 효용: 좀더 확신에 가까워지기 (쩝)
  - 마치, SW테스트에서 "간접테스트$_{\text{metamorphic test}}$" 같은

겉보기 증명$_{\text{extentional proof}}$ 스타일: 예 1

이고가기$_{\text{lifting}}$ 변환에 대해서

- 함수의 자유변수를 인자로 이고가도록 변환
- `f(x) = x+a` $\longrightarrow$ `f(x,a) = x+a`
- 함수정의를 맨바깥으로 옮기고
- 함수호출을 한상차림으로

이런 이고가기 변환은 맞는가?

- 이고가기 변환 `liftExp`:

$$\begin{aligned} \texttt{liftExp}: \quad & \texttt{(A -> X -> Y) Exp} \\ \texttt{->} \quad & \texttt{((A} \times \texttt{X) -> Y) Exp} \end{aligned}$$

- 대응하는 의미세계의 함수 lift:

$$\begin{aligned} \mathit{lift}: \quad & (A \to (X \to Y) \to (X \to Y)) \\ \to \quad & (A \times X \to Y) \to (A \times X \to Y) \end{aligned}$$

  다음 성질을 만족한다(가정):

$$(\mathit{lift} F)\ f\ (a, x) = F\ a\ (\lambda x'.f(a, x'))\ x$$

- 다음 겉모습을 증명하자: (의미세계에서 이고가기의 올바름)

$$(\mathit{lfp}(F a))\ x = (\mathit{lfp}(\mathit{lift} F))(a, x).$$

증명목표:

$$(\mathit{lfp}(F a))\ x = (\mathit{lfp}(\mathit{lift} F))(a, x)$$

고정점 인덕으로 증명: $P(\mathit{lfp}(F\ a), \mathit{lfp}(\mathit{lift}\ F))$

$$P(f, g) \stackrel{\text{let}}{=} (f\ x = g\ (a, x)).$$

- $\perp_{X \to Y}\ x = \perp_{A \times X \to Y}\ (a, x)$? 네.
- $f\ x = g\ (a, x)$이면 $P((F\ a) f, (\mathit{lift}\ F)\ g)$? 네, 왜냐면

$$\begin{array}{rcll} (\mathit{lift}\ F)\ g\ (a, x) & = & F\ a\ (\lambda x'.g(a, x'))\ x & \text{(성질)} \\ & = & F\ a\ (\lambda x'.f\ x')\ x & \text{(인덕가정)} \\ & = & F\ a\ f\ x. & \end{array}$$

겉보기 증명$_{\text{extentional proof}}$ 스타일: 예 2

마저할일전달(continuation-passing-style) 변환에 관해서.
(마저할일전달 강의 후)

딱맞는 의미구조$_{\text{full abstraction semantics}}$

무엇 의미구조$_{\text{denotational semantics}}$ 평가하기

- 정의한 의미구조가 실세계와 딱맞아야
- 무엇 의미가 같으면 실행 의미도 같고, 다르면 달라야

정의) $[\![\cdot]\!]$는 딱맞는 의미구조$_{\text{full abstraction semantics, fully abstract}}$:

$$[\![E]\!] = [\![E']\!] \Leftrightarrow \forall \text{context } C[].C[E] \overset{\text{behave}}{=} C[E'].$$

정의) E $\overset{\text{behave}}{=}$ E' : $\cdots$

- $[\![$`f(n)=if n=0 then 1 else f(n-1)`$]\!] = [\![$`g(n)=1`$]\!]$? 네. 굿.
- $[\![\infty$`||true`$]\!] = [\![$`true`$]\!]$? 음.
  - $[\![$`||`$]\!]$을 조심히 정의해야
  - 모든 경우마다 실행의미와 일치하도록

값중심언어의 무엇 의미구조denotational semantics

$$
\begin{array}{llll}
E & \to & n & \text{자연수} \\
& | & x & \text{변수} \\
& | & \texttt{fn}\, x\, E & \text{함수값} \\
& | & \texttt{rec}\, x\, E & \text{재귀값} \\
& | & E\; E & \text{함수적용}
\end{array}
$$

$$
\begin{array}{rrcl}
\text{값} & \mathbb{V} & = & \mathbb{N}_\perp + (\mathbb{V} \to \mathbb{V}) \\
\text{환경} & \mathit{Env} & = & \mathit{Var}_\perp \to \mathbb{V} \\
& [\![E]\!] & \in & \mathit{Env} \to \mathbb{V}
\end{array}
$$

$$
\begin{array}{rcl}
[\![n]\!]\, \sigma & = & n \\
[\![x]\!]\, \sigma & = & \sigma(x) \\
[\![\texttt{fn}\, x\, E]\!]\, \sigma & = & \lambda v.([\![E]\!]\sigma\{x \mapsto v\}) \\
[\![\texttt{rec}\, x\, E]\!]\, \sigma & = & \mathit{lfp}\lambda v.([\![E]\!]\sigma\{x \mapsto v\}) \\
[\![E_1\; E_2]\!]\, \sigma & = & ([\![E_1]\!]\, \sigma) \cdot ([\![E_2]\!]\, \sigma)
\end{array}
$$