SNU 4541.664A Program Analysis Spring 2005 Note 3

Prof. Kwangkeun Yi

정리:궁극을 드러내는 의미(*denotational semantics*)

- 모든 컴퓨터 프로그램의 의미(의미방정식의 해)는 의미공간 CPO의 한 원소이고, 그것은 의미방정식에 해당하는 연속함수의 최소 고정점(*least fixpoint*)으로 정의된다.
- 함수의 최소 고정점으로 정의된 프로그램의 의미에 대한 성질을 증명하는 데는 고정점 귀납법(*fixpoint induction*)이 유용하게 사용된다. 단, 그 성질이 품에넣는 성질(*inclusive assertion*)일 때에만.

- 프로그램 C의 의미

$$
\begin{array}{lcll}
[\![C]\!] & = & \mathcal{F}([\![C]\!]) & C\text{의 의미 방정식} \\
[\![C]\!] & \in & D & \\
\mathcal{F} & \in & D \to D & \\
\text{일 때,} & & & \\
[\![C]\!] & = & \mathit{fix}\mathcal{F} & C\text{의 의미} \\
 & = & \bigsqcup_{i \geq 0}(\mathcal{F}^i \bot) &
\end{array}
$$

- 의미의 성질 $P([\![C]\!])$을 증명하려면:
P가 "품에넣는 성질"일 때

$$\forall x \in \{\bot, \mathcal{F}\bot, \mathcal{F}(\mathcal{F}\bot), \cdots\}.P(x)$$

를 증명하면 됨

$$\begin{array}{rcl} M \in \mathit{Memory} & = & \mathit{Var} \rightarrow \mathit{Value} \\ z \in \mathit{Value} & = & \mathbb{Z}_\perp \\ x \in \mathit{Var} & = & \mathit{ProgramVariable}_\perp \\ [\![C]\!] & \in & \mathit{Memory} \rightarrow \mathit{Memory} \\ [\![E]\!] & \in & \mathit{Memory} \rightarrow \mathit{Value} \end{array}$$

$$\begin{array}{rcl} [\![\mathtt{skip}]\!] & = & \lambda M.M \\ [\![x \texttt{ := } E]\!] & = & \lambda M.M\{x \mapsto [\![E]\!]\ M\} \\ [\![\mathtt{if}\ E\ C_1\ C_2]\!] & = & \lambda M.\mathit{if}\ [\![E]\!]M \neq 0\ \mathit{then}\ [\![C]\!]M\ \mathit{else}\ [\![C_2]\!]M \\ [\![C_1\ ;\ C_2]\!] & = & \lambda M.[\![C_2]\!]\ ([\![C_1]\!]\ M) \\ [\![\mathtt{while}\ E\ \mathtt{do}\ C]\!] & = & \mathit{fix}(\lambda X.(\lambda M.\mathit{if}\ [\![E]\!]M \neq 0\ \mathit{then}\ X([\![C]\!]M)\ \mathit{else}\ M)) \\ [\![n]\!] & = & \lambda M.n \\ [\![E_1 \texttt{ + } E_2]\!] & = & \lambda M.([\![E_1]\!]\ M) + ([\![E_2]\!]\ M) \\ [\![\texttt{-}\ E]\!]\ M & = & \lambda M. - ([\![E]\!]\ M) \end{array}$$

소풍: 고정점(*fixpoint*) 예

"Computer science is full of fixpoints."
귀납적/재귀적으로 표현되는 것 = 최소 고정점(*least fixpoint*)을 의미:

- `N = {0} ∪ {n+1|n ∈ N}`

$$N = \mathit{fix}\lambda X.\{0\} \cup \{n+1 | n \in X\}$$

- `list = {nil} ∪ {(0,l)|l∈ list}`

$$\mathit{list} = \mathit{fix}\lambda X.\{\texttt{nil}\} \cup \{(0,l) | l \in X\}$$

- `reach(N) = N ∪ reach(next(N))`

$$reach = fix \lambda f.(\lambda N.N \cup f(next(N)))$$

- `sort(A) = if sorted(A)? A: sort(exch(A))`

$$sort = fix \lambda f.(\lambda A. if\ sorted(A) ? A : f(exch(A)))$$

- `fac(n) = if n=0? 1 : n*fac(n-1)`

  $$fac = fix \lambda f.(\lambda n.if\ n = 0?\, 1 :\ n \times f(n-1))$$

- `repeat` C E `=`
  C ; `if` E `(repeat` C E`) skip`

  $$[\![\texttt{repeat}\ C\ \ E]\!] = fix \lambda X.[\![C\ ;\ \texttt{if}\ E\ X\ \texttt{skip}]\!]$$

고정점 귀납법(*fixpoint induction*) 예

$$
\begin{array}{l}
[\![\texttt{while } E \texttt{ do } C]\!] \\
\quad = \quad \mathit{fix} \lambda X.(\lambda M.\mathit{if}\ [\![E]\!]M \neq 0\ \mathit{then}\ X([\![C]\!]M)\ \mathit{else}\ M) \\
[\![\texttt{repeat } C \ \ E]\!] \\
\quad = \quad \mathit{fix} \lambda X.(\lambda M.[\![C \texttt{ ; if } E\ X \texttt{ skip}]\!]M)
\end{array}
$$

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

$$\begin{aligned}[\![C \; ; \; \texttt{while} \; E \; \texttt{do} \; C]\!] &= (\mathit{fix} F_w) \circ [\![C]\!] \\ [\![\texttt{repeat} \;\; C \;\; E]\!] &= \mathit{fix} F_r \\ F_w, F_r &= \cdots\end{aligned}$$

증명할 것은

$$\forall C, E.(\mathit{fix} F_w) \circ [\![C]\!] = \mathit{fix} F_r$$

- 품에넣는 성질(*inclusive predicate*)? 예
- 어떤 고정점에 대한 성질? 연속함수 G의 최소고정점에 대한:

  $$G = \lambda \langle x_1, x_2 \rangle . \langle F_w x_1, F_r x_2 \rangle$$

  왜냐면

  $$\mathit{fix} G = \langle \mathit{fix} F_w, \mathit{fix} F_r \rangle$$

증명할것:

$$P(\mathit{fix}\, F_w, \mathit{fix}\, F_r) \overset{\text{let}}{=} (\forall C, E.(\mathit{fix}\, F_w) \circ [\![C]\!] = \mathit{fix}\, F_r)$$

고정점 귀납법(*fixpoint induction*)으로

- $P(\bot, \bot) \land P(F_w \bot, F_r \bot) \land P(F_w^2 \bot, F_w^2 \bot) \land \cdots$? 즉,
- $P(\bot, \bot)$ 이고 $P(f, g) \Rightarrow P(F_w f, F_r g)$ 인가?