Homework 1

SNU 4541.664A

기한: 3/25(화) 수업시간

Kwangkeun Yi

- 이번 숙제의 목적은, 귀납법 증명 연습하기, 고정점과 CPO의 개념을 확인하기, 언어의 의미구조대로 프로그램의 의미를 풀어보기, 언어의 두 가지 의미구조에 대한 성질 증명(귀납법으로) 연습하기다.

Exercise 1 식들의 집합이 귀납적으로 다음과 같이 정의된다:

$$e \to x \mid e + e \mid e * e \mid e ?\, e\, e$$

"+"와 "*"는 각각 정수 더하기와 곱하기를 뜻하고 "$e_1 ?\, e_2\, e_3$"은 e_1의 값이 0이면 e_3의 값을, 아니면 e_2의 값을 계산한다.

다음을 증명하라: 모든 식에 대해서, 그 식에 나타나는 변수들의 값이 n의 배수이면 그 식의 값은 n의 배수이다.

Exercise 2 집합 A의 부분집합들로 구성된 집합 2^A를 생각하자. 순서 $X \sqsubseteq Y$는 다음과 같이 정의된다:

$$X \sqsubseteq Y \iff X \subseteq Y.$$

CPO임을 (가장작은 원소가 무엇인지, 그리고 임의의 체인의 최소윗뚜껑(lub, least upper bound)이 항상 존재하는지)를 증명하라.

Exercise 3 CPO A에서 CPO B의 곱집합(Cartesian product)를 생각하자. 생각하자. 순서 $\langle a, b\rangle \sqsubseteq \langle a', b'\rangle$ 는 다음과 같이 정의된다:

$$\langle a, b\rangle \sqsubseteq \langle a', b'\rangle \iff a \sqsubseteq_A a' \,\wedge\, b \sqsubseteq_B b'.$$

CPO임을 (가장작은 원소가 무엇인지, 그리고 임의의 체인의 최소윗뚜껑(lub)이 항상 존재하는지)를 증명하라.

Exercise 4 CPO A에서 CPO B로 가는 연속함수로 구성된 $A \to B$를 생각하자. 순서 $f \sqsubseteq g$는 다음과 같이 정의된다:

$$f \sqsubseteq g \iff \forall a \in A : f(a) \sqsubseteq_B g(a).$$

CPO임을 (가장작은 원소가 무엇인지, 그리고 임의의 체인의 최소윗뚜껑(lub)이 항상 존재하는지)를 증명하라.

Exercise 5 다음 함수들의 고정점(fixpoint)을 찾아라:

- $\lambda x.1 \quad \in \quad \mathbb{N} \to \mathbb{N}$
- $\lambda x.x \quad \in \quad \mathbb{N} \to \mathbb{N}$
- $\lambda x.x + 1 \quad \in \quad \mathbb{N} \cup \{\infty\} \to \mathbb{N} \cup \{\infty\}$
- $\lambda X.\{\epsilon\} \cup \{\star x \mid x \in X\} \quad \in \quad 2^S \to 2^S$ (S 는 $\star$로 만든 유한 문자열들의 집합)
- $\lambda f(\lambda x.\text{if } x = 0?\ 1 :\ x \times f(x-1)) \quad \in \quad (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})$

Exercise 6 집합 S에 대해서 그 부분집합들의 집합 2^S의 원소들 사이의 부분순서 (partial order) $\sqsubseteq$를 부분집합 순서 $\subseteq$로 하면, 최소윗뚜껑(least upper bound)은 합집합과 같고 가장 작은 원소는 $\emptyset$인 CPO로 볼 수 있다.

어떤 연속 함수

$$f \in 2^S \to 2^S$$

를 생각하자. 이 함수는 다음 두 성질을 만족한다고 가정하자: $f(\emptyset) = \emptyset$ 그리고 $\forall x, y \in 2^A : f(x \cup y) = f(x) \cup f(y)$.

다음을 증명하라: $A \in 2^S$라고 하고, 최소 고정점

$$\mathit{fix}(\lambda x.A \cup f(x)) = \bigcup_{i \geq 0} f^i(A).$$

Exercise 7 다음 연속함수들의 최소 고정점(least fixpoint)을 찾아라:

- $\lambda x.1 \quad \in \quad \mathbb{N}_\perp \to \mathbb{N}_\perp$
- $\lambda x.x \quad \in \quad \mathbb{N}_\perp \to \mathbb{N}_\perp$
- $\lambda f(\lambda x.\text{if } x = 0?\ 0 :\ x + f(x-1)) \quad \in \quad (\mathbb{N}_\perp \to \mathbb{N}_\perp) \to (\mathbb{N}_\perp \to \mathbb{N}_\perp)$

  $+$함수와 $-$함수는 엄격(strict)하다. 즉, 인자가 하나라도 $\perp$이면 $\perp$을 낸다.

- $\lambda X.\{\epsilon\} \cup \{\star x \mid x \in \quad X\} \quad \in \quad 2^S \to 2^S$

  S 는 $\star$로 만드는 유한한 문자열들의 집합이다.

Exercise 8 수업시간에 다룬(교과서 2장) "그림" 언어를 생각하자.

`p`	::=	`init`$(\mathcal{R})$	initialization, with a point in $\mathcal{R}$
	|	`translation`(u, v)	translation by vector (u, v)
	|	`rotation`(u, v, θ)	rotation defined by center (u, v) and angle θ
	|	`p ; p`	sequence of operations
	|	`{p}or{p}`	choice (the branch taken is non-deterministic)
	|	`iter{p}`	iteration (the number of iterations is non-deterministic)

위 그림 언어의 의미를 수학적으로 정의하자. 다음과 같이 정의하도록 한다:

- 프로그램의 의미로 우리가 잡아내려고 하는 것은 프로그램이 점들의 집합을 어떻게 움직여 가느냐다. 실제 프로그램 실행은 하나의 점을 움직여가지만, 우리의 관심은 모든 가능한 점들의 집합들에 대한 정보다. 즉, 프로그램이 점들의 집합을 받으면 각 점들을 움직여서 어떤 점들의 집합을 만들지, 이다.

- 따라서, 그림 프로그램 `p`의 의미 $[\![\texttt{p}]\!]$는 xy-평면 점들의 집합을 받아서 xy-평면 점들의 집합을 내놓는 함수로 정의하게 된다.

- 그런 함수는 CPO위에서의 연속함수가 될 것이다. xy-평면 점들의 집합을 G 라고 하면,

$$[\![\texttt{p}]\!] \in 2^G \to 2^G.$$

- 임의의 프로그램 `p`의 의미 $[\![\texttt{p}]\!]$는 프로그램을 만드는 위의 6가지 경우마다 조립식으로 정의하면 결정된다. 다음과 같이 첫 세 가지 경우를 정의하였다:

$$\begin{array}{rcl} [\![\texttt{init}(\mathcal{R})]\!]A &=& \mathcal{R} \\ [\![\texttt{translation}(u,v)]\!]A &=& \{\mathit{trans}(p,(u,v)) \mid p \in A\} \\ [\![\texttt{rotation}(u,v,\theta)]\!]A &=& \{\mathit{rotate}(p,(u,v,\theta)) \mid p \in A\} \end{array}$$

  위에서 *trans*와 *rotate*는 정의되있다고 가정하자. 나머지 경우를 정의하라.

- 다음 프로그램 p의 의미 $[\![\texttt{p}]\!]A$를 위의 정의에 따라 써보라:

$$
\begin{array}{l}
\texttt{init}(\{(0,0),(0,1)\});\\
\texttt{iter}\{\\
\quad\{\\
\qquad\texttt{translation}(1,0);\\
\quad\}\texttt{or}\{\\
\qquad\texttt{translation}(1,1);\\
\quad\}\\
\}
\end{array}
$$

Exercise 9 한 언어의 두 가지 의미구조 사이의 성질을 증명하는 연습을 하자. 다음과 같은 단순 언어를 생각하자.

$$
\begin{array}{rcl}
C & ::= & \texttt{skip} \mid \texttt{store}\ E \mid C\ ;\ C \mid C\ \texttt{or}\ C\\
E & ::= & n \mid \texttt{load} \mid E\ \texttt{+}\ E \mid \texttt{-}\ E
\end{array}
$$

변수가 하나만 있는 명령형 언어라고 보면된다.

프로그램 C의 모듬 의미구조(collecting semantics) $[\![C]\!]$ 는 다음과 같다. 복수의 기계상태(현재 변수값)를 받아 새로운 기계상태(새로운 변수값)들을 내놓는다.

$$
\begin{array}{rcl}
Val & = & \mathbb{Z}\\
[\![C]\!] & \in & 2^{Val} \to 2^{Val}\\
[\![E]\!] & \in & 2^{Val} \to 2^{Val}
\end{array}
$$

$$
\begin{array}{rcl}
[\![\texttt{skip}]\!]S & = & S\\
[\![\texttt{store}\ E]\!]S & = & [\![E]\!]S\\
[\![C_1\ ;\ C_2]\!]S & = & [\![C_2]\!]([\![C_1]\!]S)\\
[\![C_1\ \texttt{or}\ C_1]\!]S & = & [\![C_1]\!]S\ \cup\ [\![C_2]\!]S\\
[\![n]\!]S & = & \{n\}\\
[\![\texttt{load}]\!]S & = & S\\
[\![E_1\ \texttt{+}\ E_2]\!]S & = & ([\![E_1]\!]S)\dot{+}([\![E_2]\!]S)\\
[\![\texttt{-}\ E]\!]S & = & \dot{-}([\![E]\!]S)
\end{array}
$$

$$
\begin{array}{rcl}
S_1\dot{+}S_2 & = & \{s_1 + s_2 \mid s_1 \in S_1, s_2 \in S_2\}\\
\dot{-}S & = & \{-s \mid s \in S\}
\end{array}
$$

프로그램 C의 또 다른 의미구조 $[\![C]\!]^\#$는 다음과 같이 정의된다.

$$
\begin{array}{rcl}
Val^\# & = & \{\bot, 0, 1, \top\} \\
\text{위 원소들의 의미는} & & \\
\gamma(\bot) & = & \emptyset \\
\gamma(\top) & = & \mathbb{Z} \\
\gamma(1) & = & \text{the set of all odd integers} \\
\gamma(0) & = & \text{the set of all even integers} \\
\\
[\![E]\!]^\# & \in & Val^\# \to Val^\# \\
[\![n]\!]^\# s^\# & = & n \mod 2 \\
[\![\texttt{load}]\!]^\# s^\# & = & s^\# \\
[\![E_1 \texttt{ + } E_2]\!]^\# s^\# & = & ([\![E_1]\!]^\# s^\#) +^\# ([\![E_2]\!]^\# s^\#) \\
[\![\texttt{- } E]\!]^\# s^\# & = & [\![E]\!]^\# s^\# \\
\\
\text{communtative } [\![+]\!]^\# & \text{is} & \\
\bot +^\# x & = & \bot \\
0 +^\# 0 & = & 0 \\
0 +^\# 1 & = & 1 \\
1 +^\# 1 & = & 0 \\
\top +^\# 0 & = & \top \\
\top +^\# 1 & = & \top \\
\top +^\# \top & = & \top \\
\\
[\![C]\!]^\# & \in & Val^\# \to Val^\# \\
[\![\texttt{skip}]\!]^\# s^\# & = & s^\# \\
[\![\texttt{store } E]\!]^\# s^\# & = & [\![E]\!]^\# s^\# \\
[\![C_1 \texttt{ ; } C_2]\!]^\# s^\# & = & [\![C_2]\!]^\# ([\![C_1]\!]^\# s^\#) \\
[\![C_1 \texttt{ or } C_1]\!]^\# s^\# & = & ([\![C_1]\!]^\# s^\#) \cup^\# ([\![C_2]\!]^\# s^\#) \\
\\
\text{commutative } \cup^\# & \text{is} & \\
\bot \cup^\# x & = & x \\
\top \cup^\# x & = & \top \\
0 \cup^\# 1 & = & \top \\
0 \cup^\# 0 & = & 0 \\
1 \cup^\# 1 & = & 1
\end{array}
$$

다음을 증명하라: 모든 프로그램 C 에 대해서, 의미구조 $[\![C]\!]^{\#}$는 실제 의미구조 $[\![C]\!]$를 포섭한다. 즉, $S \subseteq \gamma(s^{\#})$ 이면 $[\![C]\!]S \subseteq \gamma([\![C]\!]^{\#}s^{\#})$를 증명하라.