Program Analysis HW 6

김도형

Step 0

The semantics of D language document specifies most of the details about concrete semantics and concrete domain. We additionally define a continous function $F : \wp(\mathbb{S}) \to \wp(\mathbb{S})$,

$$F(X) = I \cup Step(X)$$

where $Step(X) = \breve{\wp}(\hookrightarrow)$.

Step 1

Abstract domain for $Value^\#, Memory^\#$ are same as HW4, hence its specifications are omitted. (To simplify, α, γ in Value, Memory domain are renamed as α_V, γ_V and α_M, γ_M, respectively.)

$$\begin{array}{rcl} \mathbb{S}^\# & = & Label \to Memory^\# \\ \hookrightarrow^\# & \subseteq & (Label \times Memory^\#) \times (Label \times Memory^\#) \end{array}$$

$$2^{\mathbb{S}} \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \mathbb{S}^\#$$

$$\begin{array}{rcl} \alpha(S) & = & \lambda l. \overset{\#}{\bigcup_{s \in S}} s(l) \\ \gamma(s^\#) & = & \{s \mid \forall l. s(l) \in \gamma_M(s^\#(l))\} \end{array}$$

Step 2

Since $Value^\#, Memory^\#$ didn't change from the HW4, its proof for Galois connection and CPO would be omitted. For $\mathbb{S}^\#$, as it is defined as $Memory^\#$ were, proving these characteristics could be done in same way. It is a CPO since it has an $\emptyset$ as a $\bot$, and any given chain would have a least upper bound attained from the least upper bound of $Memory^\#$ chain for each label. Galois connection is well-defined, since the definition of α, γ shows the Memory domain's Galois connection label-wise.

Step 3

The definition for $\hookrightarrow^\#$ can be given as:

$$\begin{aligned}
\texttt{x} := \texttt{E} &: (\ell, m^\#) \hookrightarrow^\# (\texttt{next}(\ell), \texttt{update}_x^\#(m^\#, \underline{E}^\#(m^\#))) \\
*\texttt{x} := \texttt{E} &: (\ell, m^\#) \hookrightarrow^\# (\texttt{next}(\ell), \texttt{refUpdate}_x^\#(m^\#, \underline{E}^\#(m^\#))) \\
\texttt{C}_1; \texttt{C}_2 &: (\ell, m^\#) \hookrightarrow^\# (\texttt{next}(\ell), m^\#) \\
\texttt{if E C}_1\ \texttt{C}_2 &: (\ell, m^\#) \hookrightarrow^\# (\texttt{nextTrue}(\ell), \mathcal{T}_E^\#(m^\#)) \\
&: (\ell, m^\#) \hookrightarrow^\# (\texttt{nextFalse}(\ell), \mathcal{F}_E^\#(m^\#)) \\
\texttt{while E C} &: (\ell, m^\#) \hookrightarrow^\# (\texttt{nextTrue}(\ell), \mathcal{T}_E^\#(m^\#)) \\
&: (\ell, m^\#) \hookrightarrow^\# (\texttt{nextFalse}(\ell), \mathcal{F}_E^\#(m^\#)) \\
\texttt{goto E} &: (\ell, m^\#) \hookrightarrow^\# (\ell', m^\#) \qquad (\ell' \in \gamma_V(\underline{E}^\#(m^\#)))
\end{aligned}$$

which the abstract domain's additionally defined functions are defined as HW4's $\underline{C}^\#$:

$$\begin{aligned}
\texttt{update}_x^\#(m^\#, \underline{E}^\#(m^\#)) &= \underline{x := E}^\# m^\# \\
\texttt{refUpdate}_x^\#(m^\#, \underline{E}^\#(m^\#)) &= \underline{*x := E}^\# m^\#
\end{aligned}$$

From $\hookrightarrow^\#$, we can define the abstraction for $Step, F$:

$$\begin{aligned}
F^\#(X^\#) &= \alpha(I) \cup_S^\# Step^\#(X^\#) \\
Step^\# &= \wp(\mathrm{id}, \cup^\#) \circ \pi \circ \breve{\wp}(\hookrightarrow^\#)
\end{aligned}$$

Step 4

To prove the soundness of $\hookrightarrow$, we will use the proofs in HW4 on soundness of $\underline{E}, \underline{C}$, as $\texttt{update}_x^\#, \texttt{refUpdate}_x^\#$ are same as $\underline{x := E}^\#$, $\underline{*x := E}^\#$. Only remaining problems related to $\mathbb{S}$ is trivial, since it is defined as $Memory^\#$. The soundness of $\cup_S^\#$ follows the same proof as $\cup^\#$, which joins the element in $Memory^\#$. From this, the soundness of $Step, F$ comes out consequently.

Soundness of $\hookrightarrow$

Proof of $\breve{\wp}(\hookrightarrow) \circ \gamma \subseteq \gamma \circ \breve{\wp}(\hookrightarrow^\#)$ is done by structural induction on C.

$x := E$:

$$\begin{aligned}
&(\breve{\wp}(\hookrightarrow) \circ \gamma)\{(\ell, m^\#)\} \\
=\ & \breve{\wp}(\hookrightarrow)\{(\ell, m) \mid m \in \gamma_M(m^\#)\} \\
=\ & \{(\texttt{next}(\ell), (\texttt{update}_x \circ (\mathrm{id}, \underline{E}^\#))m) \mid m \in \gamma_M(m^\#)\} \\
\subseteq\ & \{(\texttt{next}(\ell), m) \mid m \in (\wp(\texttt{update}_x) \circ \times \circ (\mathrm{id}, \wp(\underline{E}^\#)) \circ \gamma_M)m^\#\} \\
=\ & \{(\texttt{next}(\ell), m) \mid m \in (\wp(\texttt{update}_x) \circ \times \circ (\gamma_M, \wp(\underline{E}^\# \circ \gamma_M)))m^\#\} \\
\subseteq\ & \{(\texttt{next}(\ell), m) \mid m \in (\wp(\texttt{update}_x) \circ \times \circ (\gamma_M, \gamma_V \circ \wp(\underline{E}^\#)))m^\#\} \qquad (\because \text{sound } \underline{E}) \\
=\ & \{(\texttt{next}(\ell), m) \mid m \in (\wp(\texttt{update}_x) \circ \times \circ (\gamma_M, \gamma_V) \circ (\mathrm{id}, \wp(\underline{E}^\#)))m^\#\} \\
\subseteq\ & \{(\texttt{next}(\ell), m) \mid m \in (\gamma_M \circ \texttt{update}_x^\# \circ (\mathrm{id}, \wp(\underline{E}^\#)))m^\#\} \qquad (\because \text{sound } \texttt{update}) \\
=\ & \gamma\{(\texttt{next}(\ell), (\texttt{update}_x^\# \circ (\mathrm{id}, \underline{E}^\#)m^\#)\} \\
=\ & (\gamma \circ \breve{\wp}(\hookrightarrow^\#))\{(\ell, m^\#)\}
\end{aligned}$$

$*x := E$:

$$
\begin{array}{lll}
& (\breve{\wp}(\hookrightarrow) \circ \gamma)\{(\ell, m^\#)\} & \\
= & \breve{\wp}(\hookrightarrow)\{(\ell, m) \mid m \in \gamma_M(m^\#)\} & \\
= & \{(\texttt{next}(\ell), (\texttt{refUpdate}_x \circ (\mathrm{id}, \underline{E}^\#))m) \mid m \in \gamma_M(m^\#)\} & \\
\subseteq & \{(\texttt{next}(\ell), m) \mid m \in (\wp(\texttt{refUpdate}_x) \circ \times \circ (\mathrm{id}, \wp(\underline{E}^\#)) \circ \gamma_M)m^\#\} & \\
= & \{(\texttt{next}(\ell), m) \mid m \in (\wp(\texttt{refUpdate}_x) \circ \times \circ (\gamma_M, \wp(\underline{E}^\# \circ \gamma_M)))m^\#\} & \\
\subseteq & \{(\texttt{next}(\ell), m) \mid m \in (\wp(\texttt{refUpdate}_x) \circ \times \circ (\gamma_M, \gamma_V \circ \wp(\underline{E}^\#)))m^\#\} & (\because \text{sound } \underline{E}) \\
= & \{(\texttt{next}(\ell), m) \mid m \in (\wp(\texttt{refUpdate}_x) \circ \times \circ (\gamma_M, \gamma_V) \circ (\mathrm{id}, \wp(\underline{E}^\#)))m^\#\} & \\
\subseteq & \{(\texttt{next}(\ell), m) \mid m \in (\gamma_M \circ \texttt{refUpdate}^\#_x \circ (\mathrm{id}, \wp(\underline{E}^\#)))m^\#\} & (\because \text{sound } \texttt{refUpdate}) \\
= & \gamma\{(\texttt{next}(\ell), (\texttt{refUpdate}^\#_x \circ (\mathrm{id}, \underline{E}^\#)m^\#)\} & \\
= & (\gamma \circ \breve{\wp}(\hookrightarrow^\#))\{(\ell, m^\#)\} &
\end{array}
$$

$C_1; C_2$:

It is trivial since it does not change the memory, and label follows just as concrete semantics.

$\texttt{if}\ E\ C_1\ C_2$:

$$
\begin{array}{lll}
& (\breve{\wp}(\hookrightarrow) \circ \gamma)\{(\ell, m^\#)\} & \\
= & \breve{\wp}(\hookrightarrow)\{(\ell, m) \mid m \in \gamma_M(m^\#)\} & \\
= & \{(\texttt{nextTrue}(\ell), m_1) \mid m_1 \in \wp(\mathcal{T}_E)(\gamma_M(m^\#))\} \cup \{(\texttt{nextFalse}(\ell), m_2) \mid m_2 \in \wp(\mathcal{F}_E)(\gamma_M(m^\#))\} & \\
\subseteq & \{(\texttt{nextTrue}(\ell), m_1) \mid m_1 \in \gamma_M(\mathcal{T}_E(m^\#))\} \cup \{(\texttt{nextFalse}(\ell), m_2) \mid m_2 \in \gamma_M(\mathcal{F}_E(m^\#))\} & \\
& \qquad (\because \text{sound } \underline{E}) & \\
= & \gamma\{(\texttt{nextTrue}(\ell), \mathcal{T}^\#_E(m^\#), (\texttt{nextFalse}(\ell), \mathcal{F}^\#_E(m^\#)\} & \\
= & (\gamma \circ \breve{\wp}(\hookrightarrow^\#))\{(\ell, m^\#)\} &
\end{array}
$$

$\texttt{while}\ E\ C$:

Same as $\texttt{if}\ E\ C_1\ C_2$.

$\texttt{goto}\ E$:

It is trivial since it does not change the memory, and available labels are included.