Analyzing code with Separation Logic

Pfalzgraf Martin

Supervised By:
Rajeev Gore and Zhe Hou
Project

- Using a logic as a framework to reason about code
- Writing a solver to do this from scratch
Goal

- Guide the programmer through their code
- Debug Programs

- Help them gain a meaningful understanding of what their code is doing
Examples

{true} – context before execution

\[ x := 3; \]

if \( (x = 3) \)

then \{y := 2\}

else \{y := 4\};

\[ z := y + 2 \]

\{z = 4\} – context after execution
Tools

- Separation Logic as an extension of Hoare Logic
- Built an interactive solver
Hoare Logic Rules

- **Sequence:**
  \[
  \begin{align*}
  \{P\} S_1 \{Q\} & \quad \{Q\} S_2 \{R\} \\
  \{P\} S_1; S_2 \{R\}
  \end{align*}
  \]

- **Conditional:**
  \[
  \begin{align*}
  \{P \land b\} S_1 \{Q\} & \quad \{P \land \neg b\} S_2 \{Q\} \\
  \{P\} \text{ if } b \text{ then } S_1 \text{ else } S_2 \{Q\}
  \end{align*}
  \]

- **While Loop:**
  \[
  \begin{align*}
  \{P \land b\} S \{P\} \\
  \{P\} \text{ while } b \text{ do } S \{P \land \neg b\}
  \end{align*}
  \]

- **Precondition Strengthening:**
  \[
  \begin{align*}
  P_s \Rightarrow P_w & \quad \{P_w\} S \{Q\} \\
  \{P_s\} S \{Q\}
  \end{align*}
  \]

- **Postcondition Weakening:**
  \[
  \begin{align*}
  \{P\} S \{Q_s\} & \quad Q_s \Rightarrow Q_w \\
  \{P\} S \{Q_w\}
  \end{align*}
  \]

- **Assignment:**
  \[
  \{Q(e)\} x := e \{Q(x)\}
  \]

\[
\begin{align*}
\{x + 1 > 3\} & \quad x := x + 1 \quad \{x > 3\} \\
\{x > 2\} & \quad x := x + 1 \quad \{x > 3\} \\
\{x = 3\} & \quad x := x + 1 \quad \{x > 3\}
\end{align*}
\]

- **PreEq**

\[
\begin{align*}
\{x + 2 > 5\} & \quad x := x + 2 \quad \{x > 5\} \\
\{x > 3\} & \quad x := x + 2 \quad \{x > 5\}
\end{align*}
\]

- **PreStr**

\[
\begin{align*}
\{x = 3\} & \quad x := x + 1; x := x + 2 \quad \{x > 5\}
\end{align*}
\]

- **Seq**
Separation Logic Rules

Floyd Store Axiom for Separation Logic: replaces Hoare (Store) Axiom
\[
\begin{aligned}
\{x = v \land \mathsf{emp}\} & \quad x := e \quad \{x = e(v/x) \land \mathsf{emp}\} \\
& \text{where } v \text{ is an auxiliary variable which does not occur in } e
\end{aligned}
\]

Derived Floyd Store Axiom for Separation Logic:
\[
\begin{aligned}
\{\mathsf{emp}\} & \quad x := e \quad \{x = e \land \mathsf{emp}\} \\
& \text{where } x \text{ does not occur in } e
\end{aligned}
\]

Fetch Assignment Axiom
\[
\begin{aligned}
\{(x = v_1) \land (e \mapsto v_2)\} & \quad x := [s] \quad \{(x = v_2) \land (e(v_1/x) \mapsto v_2)\} \\
& \text{where } v_1 \text{ and } v_2 \text{ are auxiliary variables which do not occur in } e
\end{aligned}
\]

Derived Fetch Assignment Axiom
\[
\begin{aligned}
\{(e \mapsto v_2)\} & \quad x := [s] \quad \{(x = v_2) \land (e \mapsto v_2)\} \\
& \text{where } v_2 \text{ and } x \text{ do not occur in } e
\end{aligned}
\]

Heap Assignment Axiom
\[
\begin{aligned}
\{e \mapsto \_\} \quad [s] & \quad \{e \mapsto e_1\} \\
& \text{where } (e \mapsto \_) \text{ abbreviates } (\exists z. e \mapsto z) \text{ and } z \text{ does not occur in } e
\end{aligned}
\]

Allocation Assignment Axiom
\[
\begin{aligned}
\{x = v \land \mathsf{emp}\} & \quad x := \text{cons}(e_1, e_2, \ldots, e_n) \quad \{x \mapsto e_1(v/x), e_2(v/x), \ldots, e_n(v/x)\} \\
& \text{where } v \text{ is an auxiliary variable different from } x \text{ and not appearing in } e_1, e_2, \ldots, e_n
\end{aligned}
\]

Derived Allocation Assignment Axiom
\[
\begin{aligned}
\{\mathsf{emp}\} & \quad x := \text{cons}(e_1, e_2, \ldots, e_n) \quad \{x \mapsto e_1, e_2, \ldots, e_n\} \\
& \text{where } x \text{ does not appear in } e_1, e_2, \ldots, e_n
\end{aligned}
\]

Dispose Axiom
\[
\begin{aligned}
\{e \mapsto \_\} \quad \text{dispose}(e) \quad \{\mathsf{emp}\} \\
& \text{where } (e \mapsto \_) \text{ abbreviates } (\exists z. e \mapsto z) \text{ and } z \text{ does not occur in } e
\end{aligned}
\]

The Frame Rule:
\[
\begin{aligned}
\{P\} \quad & \text{S} \quad \{Q\} \\
\hline
\{P \cdot R\} \quad & \text{S} \quad \{Q \cdot R\} \quad \text{where no variable modified by } \text{S} \text{ appears free in } R
\end{aligned}
\]
The KeY Project