Supplement for the 3rd lecture

Department of Computer Science and Engineering
Isao Sasano

In this document we explain some supplementary explanation to Hoare triples and substitutions.

1 A derivation of the Hoare triple in p. 17

We show a derivation of the Hoare triple {true} **while** true **do** x:=1 {false} in p. 17 as follows.

$$\dfrac{\dfrac{\dfrac{\text{true} \wedge \text{true} \Rightarrow \text{true} \qquad \dfrac{}{\{\text{true}\}\ x := 1\ \{\text{true}\}}\,(\text{assign})}{\{\text{true} \wedge \text{true}\}\ x := 1\ \{\text{true}\}}\,(\text{conseq})}{\{\text{true}\}\ \textbf{while}\ \text{true}\ \textbf{do}\ x := 1\ \{\text{true} \wedge \neg\text{true}\}}\,(\text{while}) \qquad \text{true} \wedge \neg\text{true} \Rightarrow \text{false}}{\{\text{true}\}\ \textbf{while}\ \text{true}\ \textbf{do}\ x := 1\ \{\text{false}\}}\,(\text{conseq})$$

In the above derivation tree, $\{\text{true}\}\ x := 1\ \{\text{true}\}$ holds from the assignment axiom since

$$\text{true}[1/\text{x}] = \text{true}$$

holds. In the above derivation we abbreviate the assignment axiom as assign, the consequence rule as conseq, and the while rule as while.

2 Definition of substitution

Here we define substitution for logical expression, which is used in the assignment axiom. Firstly, we assume that the logical expressions used in this lecture as follows, although it is not explicitly mentioned in the slides because of the lack of space.

$$
\begin{array}{rcl}
P & := & \text{true} \mid \text{false} \\
 & \mid & P \wedge P \mid P \vee P \mid \neg P \mid P \Rightarrow P \\
 & \mid & E \leq E \mid E \geq E \mid E < E \mid E > E \mid E = E \\
E & := & N \\
 & \mid & V \\
 & \mid & E + E \\
 & \mid & E - E \\
N & := & \cdots \mid -2 \mid -1 \mid 0 \mid 1 \mid 2 \mid \cdots \\
V & := & x \mid y \mid z \mid \cdots
\end{array}
$$

This kind of definition is called *inductive definition*, which is out of scope of this lecture. We inductively define substitutions for the logical expressions defined above as follows, which is also out of scope of this lecture.

$$
\begin{array}{rcl}
P[E/x] & = & \text{case } P \text{ of} \\
 & & \quad \begin{array}{lcl}
 \text{true} & \rightarrow & \text{true} \\
 \text{false} & \rightarrow & \text{false} \\
 P_1 \wedge P_2 & \rightarrow & P_1[E/x] \wedge P_2[E/x] \\
 P_1 \vee P_2 & \rightarrow & P_1[E/x] \vee P_2[E/x] \\
 \neg P & \rightarrow & \neg P[E/x] \\
 P_1 \Rightarrow P_2 & \rightarrow & P_1[E/x] \Rightarrow P_2[E/x] \\
 E_1 \leq E_2 & \rightarrow & E_1[E/x] \leq E_2[E/x] \\
 E_1 \geq E_2 & \rightarrow & E_1[E/x] \geq E_2[E/x] \\
 E_1 < E_2 & \rightarrow & E_1[E/x] < E_2[E/x] \\
 E_1 > E_2 & \rightarrow & E_1[E/x] > E_2[E/x] \\
 E_1 = E_2 & \rightarrow & E_1[E/x] = E_2[E/x]
 \end{array} \\
E[E_0/x] & = & \text{case } E \text{ of} \\
 & & \quad \begin{array}{lcl}
 N & \rightarrow & N \\
 E_1 + E_2 & \rightarrow & E_1[E_0/x] + E_2[E_0/x] \\
 E_1 - E_2 & \rightarrow & E_1[E_0/x] - E_2[E_0/x] \\
 V & \rightarrow & \text{if } V = x \text{ then } E_0 \text{ else } V
 \end{array}
\end{array}
$$

3 Notation of Hoare triples

People use various notations for Hoare triples. In the slides we used the notation of the form $\{P_1\}\ S\ \{P_2\}$, while the original paper by Hoare [1] used the notation of the form $P_1 \; \{S\} \; P_2$, where the statements are surrounded by braces.

References

[1] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 583, 1969.