CS81 Notes: Hoare Logic

We define a simple language for programs using the following grammar:

\[
\begin{align*}
exp & \rightarrow \text{num} | \text{var} | -exp | (exp + exp) | (exp - exp) | (exp \times exp) \\
bool & \rightarrow \text{true} | \text{false} | \neg \text{bool} | (\text{bool and bool}) | (\text{bool or bool}) | (exp < exp) \\
code & \rightarrow \text{var} = \text{exp} | \text{code} \cdot \text{code} | \text{if bool then code else code end} | \text{while bool do code end}
\end{align*}
\]

A Hoare triple is a tuple consisting of \( [\alpha] C [\omega] \) where \( \alpha \) and \( \omega \) are logical formulas stating pre-condition and post-condition, respectively, and \( C \) is a program in the form specified by the above grammar.

**Proof Rules**

1. **Composition (COMP) rule:**

\[
\begin{align*}
1 & \quad [\alpha] C_1 [\beta] \\
2 & \quad [\beta] C_2 [\gamma] \\
3 & \quad [\alpha] [C_1; C_2] [\gamma]
\end{align*}
\]

2. **Conditional (COND) rule:**

\[
\begin{align*}
1 & \quad [\alpha \land \beta] C_1 [\omega] \\
2 & \quad [\alpha \land \neg \beta] C_2 [\omega] \\
3 & \quad [\alpha] [\text{if } \beta \text{ then } C_1 \text{ else } C_2 \text{ end}] [\omega]
\end{align*}
\]

3. **Partial-While (LOOP) rule:**

\[
\begin{align*}
1 & \quad [\alpha \land \beta] C [\alpha] \\
2 & \quad [\alpha] [\text{while } \beta \text{ do } C \text{ end}] [\alpha \land \neg \beta]
\end{align*}
\]

4. **Assignment (ASGN) axiom:**

\[
\begin{align*}
1 & \quad [\alpha] [E/x] [x = E] [\alpha]
\end{align*}
\]

5. **Consequence (CONS) rule:**

\[
\begin{align*}
1 & \quad [\alpha] C [\omega] \\
2 & \quad \alpha' \vdash \alpha \quad \text{pre-condition strengthening} \\
3 & \quad \omega \vdash \omega' \quad \text{post-condition weakening} \\
4 & \quad [\alpha'] C [\omega']
\end{align*}
\]
Example: Proving the correctness of the following program called FACT.

1: \( y = 1 \)
2: \textbf{while} \((x > 0)\) do
3: \quad \textbf{end} \textbf{while}
4: \textbf{end while}

The goal is to show the above program is correct. This is equivalent to proving that the following Hoare triple holds:

\[
(|x = n \land n > 0|) \text{FACT} (\|y = n!\|) \tag{1}
\]

Let \( I \) be the following logical formula

\[
I \equiv [x! \ast y = n! \land x \geq 0] \tag{2}
\]

which will be our loop invariant. To prove that FACT is correct, we prove that the following two triples hold:

\[
(|x = n \land n > 0|) [y = 1] (|I|) \tag{3}
\]

\[
(|I|) [\textbf{while} \ (x > 0) \ \textbf{do} \ y = y \ast x; \ x = x - 1 \ \textbf{end}] (\|y = n!\|) \tag{4}
\]

We proceed by showing these two triples separately:

1. Initialization:
   To show (3) holds, we may apply the ASGN rule:

\[
(|I[1/y]|) [y = 1] (|I|)
\]

But, \( I[1/y] = [x! = n! \land x \geq 0] \) and the latter is derivable from \([x = n \land n > 0]\). So, by the CONS rule (pre-condition strengthening), we get

\[
(|x = n \land n > 0|) [y = 1] (|I|)
\]

2. Loop:
   To show (4), we consider the following smaller substeps.

   (a) Body of loop:
   The goal is to show

\[
(|I \land x > 0|) [y = y \ast x; \ x = x - 1] (|I|). \tag{5}
\]

Since the body of the loop consists of a sequence of statements, we work backwards. Using ASGN, we get

\[
(|I_1|) [x = x - 1] (|I|),
\]

where \( I_1 \equiv I[(x - 1)/x] \):

\[
I_1 = [(x - 1)! \ast y = n! \land x \geq 1]. \tag{6}
\]
Now, we look at the previous statement:

\[(I_2) \ [y = y \ast x] \ (I_1).\]

where \(I_2 = I_1[(y \ast x)/y];\)

\[I_2 = [x! \ast y = n! \land x \geq 1]. \quad (7)\]

From these two assertions, we have

\[(I \land x > 0) \ [y = y \ast x] \ (I).\]

(b) Test condition of loop:

\[(I) \ [\text{while } (x > 0) \ \text{do } [y = y \ast x; \ x = x - 1] \ \text{end } (I \land x \leq 0)]\]

This follows from the LOOP rule and (5) above.

(c) Post-condition weakening:

\[(I) \ [\text{while } (x > 0) \ \text{do } [y = y \ast x; \ x = x - 1] \ \text{end } | (y = n!)]\]

This follows from the CONS rule since \((I \land x \leq 0) \vdash (y = n!).\)
Summary  The preceding arguments can be summarized by the following:

\[
\begin{align*}
(x = n \land n \geq 0) & \text{ (CONS strengthening)} \\
(x! \ast 1 = n! \land x \geq 0) & \text{ (ASGN)}
\end{align*}
\]

1:  \( y = 1 \)

\[
(x! \ast y = n! \land x \geq 0) \text{ (CONS strengthening)}
\]

2:  while \((x > 0)\) do

\[
\begin{align*}
((x - 1)! \ast (y \ast x) = n! \land x \geq 1) & \text{ (ASGN)} \\
((x - 1)! \ast y = n! \land x \geq 1) & \text{ (ASGN)}
\end{align*}
\]

3:  \( y = y \ast x \)

\[
((x - 1)! \ast y = n! \land x \geq 1) \text{ (ASGN)}
\]

4:  \( x = x - 1 \)

\[
((x! \ast y = n! \land x \geq 0) \text{ (COMP)}
\]

5:  end while

\[
\begin{align*}
((x! \ast y = n! \land x \geq 0 \land x \leq 0) & \text{ (LOOP)} \\
(y = n!) & \text{ (CONS weakening)}
\end{align*}
\]