CS81 Notes: **Hoare Logic**

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

$$
\begin{array}{rcl}
exp & \to & \mathbf{num} \mid \mathbf{var} \mid -\ exp \mid (exp + exp) \mid (exp - exp) \mid (exp * exp) \\
bool & \to & \mathbf{true} \mid \mathbf{false} \mid (\mathbf{not}\ bool) \mid (bool\ \mathbf{and}\ bool) \mid (bool\ \mathbf{or}\ bool) \mid (exp < exp) \\
code & \to & \mathbf{var} = exp \mid code\ ;\ code \mid \mathbf{if}\ bool\ \mathbf{then}\ code\ \mathbf{else}\ code\ \mathbf{end} \mid \mathbf{while}\ bool\ \mathbf{do}\ code\ \mathbf{end}
\end{array}
$$

A **Hoare triple** is a tuple consisting of $(\!|\alpha|\!)C(\!|\omega|\!)$ where α and ω 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:

$$
\begin{array}{r|l}
1 & (\!|\alpha|\!)C_1(\!|\beta|\!) \\
2 & (\!|\beta|\!)C_2(\!|\gamma|\!) \\ \hline
3 & (\!|\alpha|\!)[C_1; C_2](\!|\gamma|\!)
\end{array}
$$

2. *Conditional* (COND) rule:

$$
\begin{array}{r|l}
1 & (\!|\alpha \wedge \beta|\!)C_1(\!|\omega|\!) \\
2 & (\!|\alpha \wedge \neg\beta|\!)C_2(\!|\omega|\!) \\ \hline
3 & (\!|\alpha|\!)[\mathbf{if}\ \beta\ \mathbf{then}\ C_1\ \mathbf{else}\ C_2\ \mathbf{end}](\!|\omega|\!)
\end{array}
$$

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

$$
\begin{array}{r|l}
1 & (\!|\alpha \wedge \beta|\!)C(\!|\alpha|\!) \\ \hline
2 & (\!|\alpha|\!)[\mathbf{while}\ \beta\ \mathbf{do}\ C\ \mathbf{end}](\!|\alpha \wedge \neg\beta|\!)
\end{array}
$$

4. *Assignment* (ASGN) axiom:

$$
\begin{array}{r|l}
1 & (\!|\alpha[E/x]|\!)[x = E](\!|\alpha|\!)
\end{array}
$$

5. *Consequence* (CONS) rule:

$$
\begin{array}{r|ll}
1 & (\!|\alpha|\!)C(\!|\omega|\!) & \\
2 & \alpha' \vdash \alpha & \text{pre-condition strengthening} \\
3 & \omega \vdash \omega' & \text{post-condition weakening} \\ \hline
4 & (\!|\alpha'|\!)C(\!|\omega'|\!) &
\end{array}
$$

Example: Proving the correctness of the following program called FACT.

$y = 1$
while $(x > 0)$ **do**
  $y = y * x$;
  $x = x - 1$
end while

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

$$\llparenthesis x = n \land n > 0 \rrparenthesis \text{ FACT } \llparenthesis y = n! \rrparenthesis \tag{1}$$

Let $\mathcal{I}$ be the following logical formula

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

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

$$\llparenthesis x = n \land n > 0 \rrparenthesis \ [y = 1] \ \llparenthesis \mathcal{I} \rrparenthesis \tag{3}$$

$$\llparenthesis \mathcal{I} \rrparenthesis \ [\textbf{while } (x > 0) \textbf{ do } [y = y * x; \ x = x - 1] \textbf{ end}] \ \llparenthesis y = n! \rrparenthesis \tag{4}$$

We proceed by showing these two triples separately:

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

   $$\llparenthesis \mathcal{I}[1/y] \rrparenthesis \ [y = 1] \ \llparenthesis \mathcal{I} \rrparenthesis$$

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

   $$\llparenthesis x = n \land n > 0 \rrparenthesis \ [y = 1] \ \llparenthesis \mathcal{I} \rrparenthesis$$

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

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

   $$\llparenthesis \mathcal{I} \land x > 0 \rrparenthesis \ [y = y * x; \ x = x - 1] \ \llparenthesis \mathcal{I} \rrparenthesis. \tag{5}$$

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

   $$\llparenthesis \mathcal{I}_1 \rrparenthesis \ [x = x - 1] \ \llparenthesis \mathcal{I} \rrparenthesis,$$

   where $\mathcal{I}_1 \equiv \mathcal{I}[(x-1)/x]$:

   $$\mathcal{I}_1 = [(x-1)! * y = n! \land x \geq 1]. \tag{6}$$

Now, we look at the previous statement:

$$(\!|\mathcal{I}_2|\!)\ [y = y * x]\ (\!|\mathcal{I}_1|\!).$$

where $\mathcal{I}_2 \equiv \mathcal{I}_1[(y * x)/y]$:

$$\mathcal{I}_2 \equiv [x! * y = n! \wedge x \geq 1]. \tag{7}$$

From these two assertions, we have

$$(\!|\mathcal{I} \wedge x > 0|\!)\ [y = y * x]\ (\!|\mathcal{I}|\!).$$

(b) Test condition of loop:

$$(\!|\mathcal{I}|\!)\ [\textbf{while}\ (x > 0)\ \textbf{do}\ [y = y * x;\ x = x - 1]\ \textbf{end}\ (\!|\mathcal{I} \wedge x \leq 0|\!)$$

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

(c) Post-condition weakening:

$$(\!|\mathcal{I}|\!)\ [\textbf{while}\ (x > 0)\ \textbf{do}\ [y = y * x;\ x = x - 1]\ \textbf{end}\ ]\ (\!|y = n!|\!)$$

This follows from the CONS rule since $(\mathcal{I} \wedge x \leq 0) \vdash (y = n!)$.

Summary The preceding arguments can be summarized by the following:

$(\!| x = n \wedge n \geq 0 |\!)$ (CONS strengthening)
$(\!| x! * 1 = n! \wedge x \geq 0 |\!)$ (ASGN)
$y = 1$
$(\!| x! * y = n! \wedge x \geq 0 |\!)$ (CONS strengthening)
while $(x > 0)$ **do**
    $(\!| (x-1)! * (y * x) = n! \wedge x \geq 1 |\!)$ (ASGN)
    $y = y * x$;
    $(\!| (x-1)! * y = n! \wedge x \geq 1 |\!)$ (ASGN)
    $x = x - 1$
    $(\!| x! * y = n! \wedge x \geq 0 |\!)$ (COMP)
end while
$(\!| x! * y = n! \wedge x \geq 0 \wedge x \leq 0 |\!)$ (LOOP)
$(\!| y = n! |\!)$ (CONS weakening)