Based in part on slides by Mattox Beckman, as updated by Vikram Adve and Gul Agha
Sequencing

\[
\begin{align*}
\{P\} & \; C_1 \; \{Q\} \\
\{Q\} & \; C_2 \; \{R\} \\
\{P\} & \; C_1; \; C_2 \; \{R\}
\end{align*}
\]

**Example:**

\[
\begin{align*}
\{z = z \land z = z\} & \; x := z \; \{x = z \land z = z\} \\
\{x = z \land z = z\} & \; y := z \; \{x = z \land y = z\} \\
\{z = z \land z = z\} & \; x := z; \; y := z \; \{x = z \land y = z\}
\end{align*}
\]
Sequencing

\[
\begin{align*}
\{P\} & \quad C_1 \quad \{Q\} \quad \{Q\} \quad C_2 \quad \{R\} \\
\{P\} & \quad C_1; \quad C_2 \quad \{R\}
\end{align*}
\]

Example:

\[
\begin{align*}
\{z = z & \land z = z\} & \quad x := z \quad \{x = z & \land z = z\} \\
\{x = z & \land z = z\} & \quad y := z \quad \{x = z & \land y = z\} \\
\{z = z & \land z = z\} & \quad x := z; \quad y := z \quad \{x = z & \land y = z\}
\end{align*}
\]
Postcondition Weakening

\[
\{P\} \ C \ \{Q'\} \quad Q' \Rightarrow Q
\]

\[
\{P\} \ C \ \{Q\}
\]

Example:

\[
\{z = z \land z = z\} \quad x := z; \ y := z \quad \{x = z \land y = z\}
\]

\[
(x = z \land y = z) \Rightarrow (x = y)
\]

\[
\{z = z \land z = z\} \quad x := z; \ y := z \quad \{x = y\}
\]
Rule of Consequence

\[ P \rightarrow P' \quad \{P'\} \ C \ {Q'} \quad Q' \rightarrow Q \]

\[ \{P\} \ C \ {Q} \]

- Logically equivalent to the combination of Precondition Strengthening and Postcondition Weakening
- Uses \( P \rightarrow P' \) and \( Q' \rightarrow Q \)
If Then Else

\{P \text{ and } B\} \ C_1 \ \{Q\} \ \{P \text{ and } \neg B\} \ C_2 \ \{Q\}

\{P\} \text{ if } B \text{ then } C_1 \ \text{ else } \ C_2 \ \text{ fi} \ \{Q\}

Example: Want

\{y=a\}

\text{if } x < 0 \text{ then } y := y - x \ \text{ else } \ y := y + x \ \text{ fi}

\{y=a+|x|\}

Suffices to show:

(1) \{y=a \& x<0\} \ y := y - x \ \{y=a+|x|\} \ \text{ and}

(4) \{y=a \& \neg(x<0)\} \ y := y + x \ \{y=a+|x|\}
\{y=a&x<0\} \quad y:=y-x \quad \{y=a+|x|\}

(3) \quad (y=a&x<0) \Rightarrow y-x=a+|x| \\
(2) \quad \{y-x=a+|x|\} \quad y:=y-x \quad \{y=a+|x|\} \\
(1) \quad \{y=a&x<0\} \quad y:=y-x \quad \{y=a+|x|\}

(1) Reduces to (2) and (3) by Precondition Strengthening \\
(2) Follows from assignment axiom \\
(3) Because x<0 \Rightarrow |x| = -x
\{y = a & \neg(x < 0)\} \ y := y + x \ \{y = a + |x|\}

(6) \quad (y = a & \neg(x < 0)) \implies (y + x = a + |x|)

(5) \quad \{y + x = a + |x|\} \ y := y + x \ \{y = a + |x|\}

(4) \quad \{y = a & \neg(x < 0)\} \ y := y + x \ \{y = a + |x|\}

(4) Reduces to (5) and (6) by Precondition Strengthening

(5) Follows from assignment axiom

(6) Because \(\neg(x < 0) \implies |x| = x\)
If then else

(1) \{y=a \& x<0\} y:= y-x \{y=a+|x|\}
(4) \{y=a \& \text{not}(x<0)\} y:= y+x \{y=a+|x|\}

\begin{align*}
\{y=a\} \\
\text{if } x < 0 \text{ then } y:= y-x \text{ else } y:= y+x \\
\{y=a+|x|\}
\end{align*}

By the if_then_else rule
While

- We need a rule to be able to make assertions about *while* loops.
  - Inference rule because we can only draw conclusions if we know something about the body
  - Let’s start with:

```
{ ? } C { ? }

{ ? } while B do C od { P }
```
The loop may never be executed, so if we want $P$ to hold after, it had better hold before, so let’s try:

$$\{ ? \} \quad C \quad \{ ? \}$$

$$\{ P \} \quad \text{while} \quad B \quad \text{do} \quad C \quad \text{od} \quad \{ P \}$$
While

- If all we know is $P$ when we enter the while loop, then we all we know when we enter the body is $(P \text{ and } B)$
- If we need to know $P$ when we finish the while loop, we had better know it when we finish the loop body:

$\{P \text{ and } B\} \ C \ \{P\} \\{P\} \text{ while } B \text{ do } C \text{ od} \ {P}$
We can strengthen the previous rule because we also know that when the loop is finished, \( \text{not } B \) also holds.

Final **while** rule:

\[
\{ P \text{ and } B \} \quad C \quad \{ P \} \\
\{ P \} \quad \text{while } B \quad \text{do } \quad C \quad \text{od} \quad \{ P \text{ and not } B \}
\]
While

\[
\{ P \text{ and } B \} \ C \ \{ P \}
\]

\[
\{ P \} \text{ while } B \ \text{ do } C \ \text{ od} \ \{ P \text{ and not } B \}
\]

- P satisfying this rule is called a loop invariant because it must hold before and after the each iteration of the loop.
While rule generally needs to be used together with precondition strengthening and postcondition weakening.

There is NO algorithm for computing the correct $P$; it requires intuition and an understanding of why the program works.
Example

Let us prove

\{x \geq 0 \text{ and } x = a\}

\text{fact} := 1;

\text{while } x > 0 \text{ do (fact := fact } \times \text{ x; x := x } - 1) \text{ od}

\{\text{fact} = a!\}
Example

- We need to find a condition $P$ that is true both before and after the loop is executed, and such that

$$(P \text{ and not } x > 0) \implies (\text{fact} = a!)$$
Example

- First attempt:
  \{a! = \text{fact} \ast (x!)\}

- Motivation:
- What we want to compute: \( a! \)
- What we have computed: \( \text{fact} \)
  which is the sequential product of \( a \) down through \( (x + 1) \)
- What we still need to compute: \( x! \)
Example

By post-condition weakening suffices to show
1. \( \{ x \geq 0 \text{ and } x = a \} \)
   
   \[
   \text{fact} := 1; \\
   \text{while } x > 0 \text{ do (fact := fact } * x; x := x - 1) \text{ od} \\
   \{ a! = \text{fact } * (x!) \text{ and not } (x > 0) \}
   \]

   and

2. \( \{ a! = \text{fact } * (x!) \text{ and not } (x > 0) \} \implies \{ \text{fact} = a! \} \)
Problem

2. \{a! = \text{fact} * (x!) \text{ and not } (x > 0)\} \rightarrow \{\text{fact} = a!\}
- Don’t know this if \(x < 0\)
- Need to know that \(x = 0\) when loop terminates
- Need a new loop invariant
- Try adding \(x \geq 0\)
- Then will have \(x = 0\) when loop is done
Example

Second try, combine the two:

\[ P = \{a! = fact \times (x!) \text{ and } x \geq 0\} \]

Again, suffices to show
1. \( \{x \geq 0 \text{ and } x = a\} \)
   
   \[
   \begin{align*}
   \text{fact} & := 1; \\
   \text{while } x > 0 \text{ do (fact} & := \text{fact} \times x; x := x - 1) \text{ od} \\
   \{P \text{ and not } x > 0\}
   \end{align*}
   \]

and

2. \( \{P \text{ and not } x > 0\} \Rightarrow \{\text{fact} = a!\} \)
Example

- For 2, we need

\[\{a! = \text{fact} \times (x!) \text{ and } x \geq 0 \text{ and not } (x > 0)\} \Rightarrow \{\text{fact} = a!\}\]

But \(\{x \geq 0 \text{ and not } (x > 0)\}\) \(\Rightarrow\) \(\{x = 0\}\) so

\[\text{fact} \times (x!) = \text{fact} \times (0!) = \text{fact}\]

Therefore

\[\{a! = \text{fact} \times (x!) \text{ and } x \geq 0 \text{ and not } (x > 0)\} \Rightarrow \{\text{fact} = a!\}\]
Example

For 1, by the sequencing rule it suffices to show

3. \( \{x \geq 0 \text{ and } x = a\}\)
   
   \[
   \text{ fact } := 1
   \]
   
   \[
   \{a! = \text{fact} \times (x!) \text{ and } x \geq 0\}\]

And

4. \( \{a! = \text{fact} \times (x!) \text{ and } x \geq 0\}\)
   
   \[
   \text{while } x > 0 \text{ do}
   \]
   
   \[
   (\text{fact } := \text{fact} \times x; \ x := x - 1) \text{ od}
   \]
   
   \[
   \{a! = \text{fact} \times (x!) \text{ and } x \geq 0 \text{ and not } (x > 0)\}\]
Example

Suffices to show that

\{a! = \text{fact} \times (x!) \text{ and } x \geq 0\}

holds before the while loop is entered and that if

\{(a! = \text{fact} \times (x!)) \text{ and } x \geq 0 \text{ and } x > 0\}

holds before we execute the body of the loop, then

\{(a! = \text{fact} \times (x!)) \text{ and } x \geq 0\}

holds after we execute the body
Example

By the assignment rule, we have

\{a! = 1 \times (x!) \text{ and } x \geq 0\}

\text{fact} := 1

\{a! = \text{fact} \times (x!) \text{ and } x \geq 0\}

Therefore, to show (3), by
precondition strengthening, it suffices
to show

\((x \geq 0 \text{ and } x = a) \Rightarrow (a! = 1 \times (x!) \text{ and } x \geq 0)\)
Example

\[(x \geq 0 \text{ and } x = a) \Rightarrow (a! = 1 \times (x!) \text{ and } x \geq 0)\]

holds because \(x = a \Rightarrow x! = a!\)

Have that \(\{a! = \text{fact} \times (x!) \text{ and } x \geq 0\}\)
holds at the start of the while loop
Example

To show (4):

\{a! = \text{fact} \times (x!) \text{ and } x \geq 0\}
while \ x > 0 \ do
(fact := fact \times x; \ x := x - 1)
\od
\{a! = \text{fact} \times (x!) \text{ and } x \geq 0 \text{ and not } (x > 0)\}

we need to show that

\{(a! = \text{fact} \times (x!)) \text{ and } x \geq 0\}

is a loop invariant
Example

We need to show:

\{(a! = fact * (x!)) \land x \geq 0 \land x > 0\}

( fact = fact * x; x := x − 1 )

\{(a! = fact * (x!)) \land x \geq 0\}

We will use assignment rule, sequencing rule and precondition strengthening
Example

By the assignment rule, we have
\{(a! = \text{fact} * ((x-1)!)) \text{ and } x - 1 \geq 0\}
\[x := x - 1\]
\{(a! = \text{fact} * (x!)) \text{ and } x \geq 0\}

By the sequencing rule, it suffices to show
\{(a! = \text{fact} * (x!)) \text{ and } x \geq 0 \text{ and } x > 0\}
\text{fact} = \text{fact} * x
\{(a! = \text{fact} * ((x-1)!)) \text{ and } x - 1 \geq 0\}
Example

By the assignment rule, we have that
\[(a! = (\text{fact} \times x) \times ((x-1)!)) \text{ and } x - 1 \geq 0\]
\[\text{fact} = \text{fact} \times x\]
\[(a! = \text{fact} \times ((x-1)!)) \text{ and } x - 1 \geq 0\]

By Precondition strengthening, it suffices to show that
\[(a! = \text{fact} \times (x!)) \text{ and } x \geq 0 \text{ and } x > 0 \implies (a! = (\text{fact} \times x) \times ((x-1)!)) \text{ and } x - 1 \geq 0)\]
Example

However

\[ \text{fact} \times x \times (x - 1)! = \text{fact} \times (x!) \]

and \((x > 0) \Rightarrow x - 1 \geq 0\)

since \(x\) is an integer, so

\[
\{ (a! = \text{fact} \times (x!)) \text{ and } x \geq 0 \text{ and } x > 0 \} \Rightarrow \\
\{ (a! = (\text{fact} \times x) \times ((x-1)!)) \text{ and } x - 1 \geq 0 \} \]
Example

Therefore, by precondition strengthening

\{(a! = fact * (x!)) \text{ and } x \geq 0 \text{ and } x > 0\}

\quad \text{fact} = \text{fact} * x

\{(a! = fact * ((x-1)!)) \text{ and } x - 1 \geq 0\}

This finishes the proof