Starter Questions

Feel free to discuss these with your neighbour:

- Consider two states $s_1$ and $s_2$ such that
  \[
  \langle s_1, x := x + 1 \rangle \Rightarrow s_2
  \]
  If predicate $P \equiv (x = y + 1)$ is true for $s_2$ then what does that tell us about $s_1$? How can we change $P$ so that it is true for $s_1$?

- What happens to the value of the expression $x + y$ every time we execute this loop?
  \[
  \text{while } x > 0 \text{ do } (y := y + z; x := x := x - z)
  \]
- What is the logical relationship between $x > 0$ and $x \geq 0$?
- What is the logical relationship between $x > y$ and $x \neq y \land x \geq y$?
Lecture 4
Examples of Proving Partial Correctness
COMP11212

Giles Reger

March 2017
This Hoare triple specifies program $S$

$$\{ P \} S \{ Q \}$$

where $P$ is a precondition and $Q$ is a postcondition.

The meaning of this is that if $P(s_{start})$ holds then $Q(s_{end})$ should hold.
Axiomatic System

\text{ass}_p \quad \{ P[x \mapsto A[a]] \} \ x := a \ { P \}

\text{skip}_p \quad \{ P \} \ \text{skip} \ { P \}

\text{comp}_p \quad \begin{align*}
\{ P \} \ S_1 \ { Q \}, \ { Q \} \ S_2 \ { R \} \\
\{ P \} \ S_1; \ S_2 \ { R \}
\end{align*}

\text{if}_p \quad \begin{align*}
\{ P \land B[b] \} \ S_1 \ { Q \}, \ { P \land \neg B[b] \} \ S_2 \ { Q \} \\
\{ P \} \ \text{if} \ b \ \text{then} \ S_1 \ \text{else} \ S_2 \ { Q \}
\end{align*}

\text{while}_p \quad \begin{align*}
\{ P \land B[b] \} \ S \ { P \} \\
\{ P \} \ \text{while} \ b \ \text{do} \ S \ { P \land \neg B[b] \}
\end{align*}
The obvious ones

Skipping doesn’t change anything

\[
\{ P \} \text{ skip } \{ P \}
\]

For sequence \( S_1; S_2 \), the postcondition of \( S_1 \) needs to make the precondition of \( S_2 \) true

\[
\begin{align*}
\{ P \} & S_1 \{ Q \}, \{ Q \} S_2 \{ R \} \\
\{ P \} & S_1; S_2 \{ R \}
\end{align*}
\]

For conditionals we can assume the condition and check the statement

\[
\begin{align*}
\{ P \land B[b] \} & S_1 \{ Q \}, \{ P \land \neg B[b] \} S_2 \{ Q \} \\
\{ P \} & \text{ if } b \text{ then } S_1 \text{ else } S_2 \{ Q \}
\end{align*}
\]
People often find this rule counterintuitive

\[
\{ P[x \mapsto A[a]] \} \ x := a \ \{ P \} 
\]

as it feels backwards

The handout discusses alternatives and why they don’t work

The justification for this approach was given in the starter question. Consider two states \( s_1 \) and \( s_2 \) such that

\[
\langle s_1, x := a \rangle \Rightarrow s_2
\]

We want to find a predicate that is true for \( s_1 \). The states \( s_1 \) and \( s_2 \) only differ in the value of \( x \). If \( P \) is true for \( s_2 \) then we know it is true when \( x \) is replaced by the value for \( a \).
Motivating The Rule of Consequence

How should we prove

\[ \{ \text{true} \} \ x := 1 \ \{ \ x > 0 \} \]

Applying assignment backwards from \( x > 0 \) gives us

\[ \{ \ 1 > 0 \} \ x := 1 \ \{ \ x > 0 \} \]

and we know ? so that seems okay. But what about

\[ \{ \ x > 1 \} \ x := x - 1 \ \{ \ x \geq 0 \} \]

again going backwards gives us

\[ \{ \ x - 1 \geq 0 \equiv x \geq 1 \} \ x := x - 1 \ \{ \ x \geq 0 \} \]

this time we know ? so it works
Motivating The Rule of Consequence

How should we prove

\[
\{ \text{true} \} \ x := 1 \ \{ \ x > 0 \ \}
\]

Applying assignment backwards from \( x > 0 \) gives us

\[
\{ \ 1 > 0 \ \} \ x := 1 \ \{ \ x > 0 \ \}
\]

and we know \( 1 > 0 \equiv \text{true} \) so that seems okay. But what about

\[
\{ \ x > 1 \ \} \ x := x - 1 \ \{ \ x \geq 0 \ \}
\]

again going backwards gives us

\[
\{ \ x - 1 \geq 0 \equiv x \geq 1 \ \} \ x := x - 1 \ \{ \ x \geq 0 \ \}
\]

this time we know \( ? \) so it works
How should we prove

\[
\begin{align*}
\{ \text{true} \} \ &x := 1 \ &\{ x > 0 \} \\
\end{align*}
\]

Applying assignment backwards from \(x > 0\) gives us

\[
\begin{align*}
\{ 1 > 0 \} \ &x := 1 \ &\{ x > 0 \} \\
\end{align*}
\]

and we know \(1 > 0 \equiv \text{true}\) so that seems okay. But what about

\[
\begin{align*}
\{ x > 1 \} \ &x := x - 1 \ &\{ x \geq 0 \} \\
\end{align*}
\]

again going backwards gives us

\[
\begin{align*}
\{ x - 1 \geq 0 \equiv x \geq 1 \} \ &x := x - 1 \ &\{ x \geq 0 \} \\
\end{align*}
\]

this time we know \(x > 1 \rightarrow x \geq 1\) so it works
The Rule of Consequence

The above slide missed a very important rule

\[
\text{cons}_p \left\{ \begin{array}{c} P' \\ \{ P \} \end{array} \right\} S \left\{ \begin{array}{c} Q' \\ \{ Q \} \end{array} \right\} \quad \text{if} \quad P \to P' \quad \text{and} \quad Q' \to Q
\]

This says that we can

- **strengthen** the precondition; and
- **weaken** the postcondition

A predicate \( A \) is **stronger** than \( B \) if it holds for **more** inputs

Another way of saying that \( A \) is (non strictly) weaker than \( B \) is \( A \to B \) as this states that \( B \) is true for as many inputs as \( A \) (but maybe more)
For each pair of logical statements which is **stronger**? Are they **equivalent**?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = z$</td>
<td>$z = x$</td>
<td></td>
</tr>
<tr>
<td>$x &gt; 0$</td>
<td>$x \geq 0$</td>
<td></td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td>$x \geq y + 1$</td>
<td></td>
</tr>
<tr>
<td>$p \land q$</td>
<td>$p$</td>
<td></td>
</tr>
<tr>
<td>$p \lor q$</td>
<td>$q$</td>
<td></td>
</tr>
<tr>
<td>$x &gt; a + b \land b &gt; 0$</td>
<td>$x &gt; a$</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>x</td>
<td>= a \land x &lt; 0$</td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td>$x \leq y$</td>
<td></td>
</tr>
</tbody>
</table>
Quiz: Weaker or Stronger?

For each pair of logical statements which is **stronger**? Are they **equivalent**?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x = z )</td>
<td>( z = x )</td>
<td>equivalent</td>
</tr>
<tr>
<td>( x &gt; 0 )</td>
<td>( x \geq 0 )</td>
<td></td>
</tr>
<tr>
<td>( x &gt; y )</td>
<td>( x \geq y + 1 )</td>
<td></td>
</tr>
<tr>
<td>( p \land q )</td>
<td>( p )</td>
<td></td>
</tr>
<tr>
<td>( p \lor q )</td>
<td>( q )</td>
<td></td>
</tr>
<tr>
<td>( x &gt; a + b \land b &gt; 0 )</td>
<td>( x &gt; a )</td>
<td></td>
</tr>
<tr>
<td>(</td>
<td>x</td>
<td>= a \land x &lt; 0 )</td>
</tr>
<tr>
<td>( x &gt; y )</td>
<td>( x \leq y )</td>
<td></td>
</tr>
</tbody>
</table>
### Quiz: Weaker or Stronger?

For each pair of logical statements which is stronger? Are they equivalent?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x = z)</td>
<td>(z = x)</td>
<td>equivalent</td>
</tr>
<tr>
<td>(x &gt; 0)</td>
<td>(x \geq 0)</td>
<td>(x \geq 0)</td>
</tr>
<tr>
<td>(x &gt; y)</td>
<td>(x \geq y + 1)</td>
<td></td>
</tr>
<tr>
<td>(p \land q)</td>
<td>(p)</td>
<td></td>
</tr>
<tr>
<td>(p \lor q)</td>
<td>(q)</td>
<td></td>
</tr>
<tr>
<td>(x &gt; a + b \land b &gt; 0)</td>
<td>(x &gt; a)</td>
<td></td>
</tr>
<tr>
<td>(</td>
<td>x</td>
<td>= a \land x &lt; 0)</td>
</tr>
<tr>
<td>(x &gt; y)</td>
<td>(x \leq y)</td>
<td></td>
</tr>
</tbody>
</table>
For each pair of logical statements which is **stronger**? Are they **equivalent**?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x = z )</td>
<td>( z = x )</td>
<td>equivalent</td>
</tr>
<tr>
<td>( x &gt; 0 )</td>
<td>( x \geq 0 )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( x &gt; y )</td>
<td>( x \geq y + 1 )</td>
<td>equivalent</td>
</tr>
<tr>
<td>( p \land q )</td>
<td>( p )</td>
<td></td>
</tr>
<tr>
<td>( p \lor q )</td>
<td>( q )</td>
<td></td>
</tr>
<tr>
<td>( x &gt; a + b \land b &gt; 0 )</td>
<td>( x &gt; a )</td>
<td></td>
</tr>
<tr>
<td>(</td>
<td>x</td>
<td>= a \land x &lt; 0 )</td>
</tr>
<tr>
<td>( x &gt; y )</td>
<td>( x \leq y )</td>
<td></td>
</tr>
</tbody>
</table>
For each pair of logical statements which is **stronger**? Are they **equivalent**?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = z$</td>
<td>$z = x$</td>
<td>equivalent</td>
</tr>
<tr>
<td>$x &gt; 0$</td>
<td>$x \geq 0$</td>
<td>$x \geq 0$ equivalent</td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td>$x \geq y + 1$</td>
<td></td>
</tr>
<tr>
<td>$p \land q$</td>
<td>$p$</td>
<td>$p$</td>
</tr>
<tr>
<td>$p \lor q$</td>
<td>$q$</td>
<td></td>
</tr>
<tr>
<td>$x &gt; a + b \land b &gt; 0$</td>
<td>$x &gt; a$</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>x</td>
<td>= a \land x &lt; 0$</td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td>$x \leq y$</td>
<td></td>
</tr>
</tbody>
</table>
Quiz: Weaker or Stronger?

For each pair of logical statements which is stronger? Are they equivalent?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = z$</td>
<td>$z = x$</td>
<td>equivalent</td>
</tr>
<tr>
<td>$x &gt; 0$</td>
<td>$x \geq 0$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td>$x \geq y + 1$</td>
<td>equivalent</td>
</tr>
<tr>
<td>$p \land q$</td>
<td>$p$</td>
<td>$p$</td>
</tr>
<tr>
<td>$p \lor q$</td>
<td>$q$</td>
<td>$p \lor q$</td>
</tr>
<tr>
<td>$x &gt; a + b \land b &gt; 0$</td>
<td>$x &gt; a$</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>x</td>
<td>= a \land x &lt; 0$</td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td>$x \leq y$</td>
<td></td>
</tr>
</tbody>
</table>
Quiz: Weaker or Stronger?

For each pair of logical statements which is stronger? Are they equivalent?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x = z )</td>
<td>( z = x )</td>
<td>equivalent</td>
</tr>
<tr>
<td>( x &gt; 0 )</td>
<td>( x \geq 0 )</td>
<td>( x \geq 0 )</td>
</tr>
<tr>
<td>( x &gt; y )</td>
<td>( x \geq y + 1 )</td>
<td>equivalent</td>
</tr>
<tr>
<td>( p \land q )</td>
<td>( p )</td>
<td>( p )</td>
</tr>
<tr>
<td>( p \lor q )</td>
<td>( q )</td>
<td>( p \lor q )</td>
</tr>
<tr>
<td>( x &gt; a + b \land b &gt; 0 )</td>
<td>( x &gt; a )</td>
<td>( x &gt; a )</td>
</tr>
<tr>
<td>(</td>
<td>x</td>
<td>= a \land x &lt; 0 )</td>
</tr>
<tr>
<td>( x &gt; y )</td>
<td>( x \leq y )</td>
<td></td>
</tr>
</tbody>
</table>
Quiz: Weaker or Stronger?

For each pair of logical statements which is stronger? Are they equivalent?

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>x = z</td>
<td>z = x</td>
<td>equivalent</td>
<td></td>
</tr>
<tr>
<td>x &gt; 0</td>
<td>x ≥ 0</td>
<td>x ≥ 0</td>
<td></td>
</tr>
<tr>
<td>x &gt; y</td>
<td>x ≥ y + 1</td>
<td>equivalent</td>
<td></td>
</tr>
<tr>
<td>p ∧ q</td>
<td>p</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>p ∨ q</td>
<td>q</td>
<td>p ∨ q</td>
<td></td>
</tr>
<tr>
<td>x &gt; a + b ∧ b &gt; 0</td>
<td>x &gt; a</td>
<td>x &gt; a</td>
<td></td>
</tr>
<tr>
<td></td>
<td>x &lt; 0</td>
<td>−x = a</td>
<td></td>
</tr>
<tr>
<td></td>
<td>x &gt; y</td>
<td>x ≤ y</td>
<td></td>
</tr>
</tbody>
</table>
Quiz: Weaker or Stronger?

For each pair of logical statements which is **stronger**? Are they **equivalent**?

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = z$</td>
<td>$z = x$</td>
<td>equivalent</td>
</tr>
<tr>
<td>$x &gt; 0$</td>
<td>$x \geq 0$</td>
<td>$x \geq 0$</td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td>$x \geq y + 1$</td>
<td>equivalent</td>
</tr>
<tr>
<td>$p \land q$</td>
<td>$p$</td>
<td>$p$</td>
</tr>
<tr>
<td>$p \lor q$</td>
<td>$q$</td>
<td>$p \lor q$</td>
</tr>
<tr>
<td>$x &gt; a + b \land b &gt; 0$</td>
<td>$x &gt; a$</td>
<td>$x &gt; a$</td>
</tr>
<tr>
<td>$</td>
<td>x</td>
<td>= a \land x &lt; 0$</td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td>$x \leq y$</td>
<td>inconsistent</td>
</tr>
</tbody>
</table>

The strong relation is a partial order.
A Picture to Motivate the Rule of Consequence

\[ P \rightarrow P' \text{ and } Q' \rightarrow Q \]
A Picture to Motivate the Rule of Consequence

\[ P \rightarrow P' \text{ and } Q' \rightarrow Q \]
while Loops

The rule for while loops is

\[
\{ P \land B[b] \} \quad S \quad \{ P \}
\]

\[
\{ P \} \quad \text{while } b \quad \text{do} \quad S \quad \{ P \land \neg B[b] \}
\]

where \( P \) is referred to as a loop invariant

We motivated this informally before. Now let’s see it in action on the division program!
Total Correctness: An Informal Argument

Does this program always terminate?

\[
\text{if } x > y \ \text{then } z := x \ \text{else} \ z := y
\]
Does this program always terminate?

if $x > y$ then $z := x$ else $z := y$

Programs without loops will always result in a state in a finite number of rewritings using $\Rightarrow$. We can prove this as a property of the programming language.
Does this program always terminate?

\[
\begin{align*}
  r &:= x; \\
  d &:= 0; \\
  \text{while } y \leq r \text{ do } (d := d+1; \ r := r-y)
\end{align*}
\]
Does this program always terminate?

\[
\begin{align*}
    r & := x; \\
    d & := 0; \\
    \textbf{while } y \leq r \textbf{ do } (d := d+1; \ r := r-y)
\end{align*}
\]

Why? When does it terminate?
Does this program always terminate?

\[
\begin{align*}
r & := x; \\
d & := 0; \\
\textbf{while } y \leq r \textbf{ do } (d := d+1; \ r := r - y)
\end{align*}
\]

Why? When does it terminate? When \( y > 0 \)
Does this program always terminate?

\[
\begin{align*}
    r &:= x; \\
    d &:= 0; \\
    \textbf{while } y \leq r \textbf{ do } (d := d+1; \ r := r - y)
\end{align*}
\]

Why? When does it terminate? When \( y > 0 \)

If \( y > 0 \) then the value of \( r \) decreases every time we go round the loop.

The loop terminates when \( y > r \) i.e. when \( r \) becomes smaller than \( y \) (as the value of \( y \) isn’t changing).

Whatever the size of \( r \) at the start, we can only decrease it a finite number of times before it is less than \( y \).
Total Correctness: The Formal Bit

Only one thing changes, the rule for while

\[
\frac{[P \land B[b] \land E = n] \quad C [P \land E < n]}{[P \land B[b] \land E = n] \quad C [P \land E < n] \quad \text{if } P \land B[b] \rightarrow E \geq 0}
\]

Here \( E \) is a loop variant i.e. an expression that decreases on every iteration. It is important to note that the loop condition needs to bound this expression from below.

It is not guaranteed that we can always find a loop variant for a terminating loop.

Conceptually, \( E \) is the maximum number of loop iterations left