Do you remember?

- A **Hoare triple** is \{ P \} S \{ Q \} and is true if whenever we run program S on a state satisfying P any resulting state satisfies Q.
- This is **partial correctness** when we also require termination it is **total correctness** and we write [ P ] S [ Q ]
- We often need **auxiliary variables** in specifications
- We have a set of axiomatic rules for proving (partial) correctness

Something to think about:

- If \( x = a \) and \( y = b \) before running this program

  ```
  while \( x > 0 \) do \\
  x := x - 1; \ y := y + 1
  ```

  what can we say about \( x \) and \( y \) after executing the loop once, twice, \( x - 1 \) times, until the loop completes?
We introducing the following notion of a specification of program $S$

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

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

We often supply $P$ and $Q$ as a specification for some $S$.

Some trivial examples of valid triples:

- $\{ x = 1 \} x := x + 1 \{ x = 2 \}$
- $\{ x > 0 \} x := x + 1 \{ x > 0 \}$
- $\{ x \neq y \} z := x \{ z \neq y \}$
- $\{ x = a \land y = b \} t := x; x := y; y := t \{ x = b \land y = a \}$
**Axiomatic System**

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

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

\[
\text{comp}_p \quad \frac{\{ P \} \quad S_1 \quad \{ Q \}, \quad \{ Q \} \quad S_2 \quad \{ R \}}{\{ P \} \quad S_1; \quad S_2 \quad \{ R \}}
\]

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

\[
\text{while}_p \quad \frac{\{ P \land B[b] \} \quad S \quad \{ P \}}{\{ P \} \quad \text{while } b \text{ do } S \quad \{ P \land \neg B[b] \}}
\]
What does $P[x \mapsto A[a]]$ mean?

It can be read as ‘replace all occurrences of $x$ in $P$ with $a$ (after evaluation)’

For example:
- $(x + y = z)[x \mapsto 5]$ becomes $5 + y = z$
- $(x > 0)[x \mapsto x + 1]$ becomes $x + 1 > 0$
- $(x \neq y)[y \mapsto x + 1 - 1]$ becomes $x \neq x$
A couple of assignment examples

\begin{equation*}
\{ P[x \mapsto A[a]] \} \ x := a \ \{ P \}
\end{equation*}

\begin{equation*}
\begin{aligned}
\{ & z := z + 1 \ \\
& x := y \ \\
\} \quad \{ z = 2 \}
\end{aligned}
\end{equation*}
A couple of assignment examples

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

\[ \{ \ (z = 2)[z \mapsto z + 1] \ \} \ \underbrace{z}_{x} := \underbrace{z + 1}_{a} \ \{ \underbrace{z = 2}_{P} \ \} \]

\[ \{ \ \} \ \underbrace{x}_{x} := \underbrace{y}_{a} \ \{ \underbrace{x = y}_{P} \ \} \]
A couple of assignment examples

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

\{ z + 1 = 2 \} \ z := z + 1 \ { z = 2 \}

\{ \} \ x := y \ { x = y \}

Giles Reger
Lecture 1
March 2017
A couple of assignment examples

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

{ $z = 1$ } $z := z + 1$ { $z = 2$ }

{ $x := y$ } $x \mapsto a$ { $x = y$ }
A couple of assignment examples

\{ P[x \mapsto A[a]] \} \ x := a \ \{ P \} \\
\{ z = 1 \} \ z := z + 1 \ \{ z = 2 \} \\
\{ (x = y)[x \mapsto y] \} \ x := y \ \{ x = y \}
A couple of assignment examples

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

\[
\{ \ z = 1 \ \} \ z := z + 1 \ \{ \ z = 2 \ \}
\]

\[
\{ \ y = y \ \} \ x := y \ \{ \ x = y \ \}
\]
A couple of assignment examples

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

\{ z = 1 \} \ z := z + 1 \ \{ z = 2 \}

\{ \} \ x := y \ \{ x = y \}
Some Examples

\{ x \geq 0 \} \ x := x + 1 \ \{ x > 0 \}

\{ \ \} \ x := 0; y := 0 \ \{ x = y \}

\text{if } y > x

\{ x \neq y \} \ \text{then } t := y; y := x; x := t \ \{ x > y \land x \neq y \}

\text{else skip}

\{ x = a \land y = b \} \ z := x; z := z + y \ \{ z = a + b \}
Some Examples

\{ \ x \geq 0 \ \} \ x := x + 1 \ \{ \ x > 0 \ \}

\{ \ \} \ x := 0; y := 0 \ \{ \ x = y \ \}

if \ y > x
\{ \ x \neq y \ \} \ \text{then} \ t := y; y := x; x := t \ \{ \ x > y \land x \neq y \ \}
\text{else skip}

\{ \ x = a \land y = b \ \} \ z := x; z := z + y \ \{ \ z = a + b \ \}

Is it always the case that \ x = a \land y = b \ is the same as \ a + b = x + y \?
Some Examples

\{ x \geq 0 \} \ x := x + 1 \ \{ x > 0 \}

\{ \} \ x := 0; \ y := 0 \ \{ x = y \}

if \ y > x
\{ x \neq y \} \ \text{then} \ t := y; \ y := x; \ x := t \ \{ x > y \land x \neq y \}
else \text{skip}

\{ x = a \land y = b \} \ z := x; \ z := z + y \ \{ z = a + b \}

Is it always the case that $x = a \land y = b$ is the same as $a + b = x + y$?

$(x = a \land y = b) \to (a + b = x + y)$

$(a + b = x + y) \not\leftrightarrow (x = a \land y = b)$ e.g. $[x \mapsto 0, y \mapsto 1, a \mapsto 1, b \mapsto 0]$
Motivating The Rule of Consequence (Again)

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 (why shortly)
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 (why shortly)
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 $x > 1 \rightarrow x \geq 1$ so it works (why shortly)
The Rule of Consequence

The above slide missed a very important rule

\[
\text{cons}_P \begin{array}{c} \{ P' \} \ S \ \{ Q' \} \\ \{ P \} \ S \ \{ Q \} \end{array} \quad \text{if } P \rightarrow P' \text{ and } Q' \rightarrow Q
\]

This says that we can

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

A predicate \( A \) is **stronger** than \( B \) if \( A \rightarrow B \) i.e. anything that follows from \( B \) also follows from \( A \) (see circles next).

Page 77 of the notes has been corrected
A Picture to Motivate the Rule of Consequence

\[ \text{cons}_p \begin{array}{c} \{ P' \} \ S \ \{ Q' \} \\ \{ P \} \ S \ \{ Q \} \end{array} \quad \text{if } P \rightarrow P' \text{ and } Q' \rightarrow Q \]
A Picture to Motivate the Rule of Consequence

\[
\text{cons}_P \quad \begin{array}{c} \{ P' \} \quad S \quad \{ Q' \} \\ \{ P \} \quad S \quad \{ Q \} \end{array} \quad \text{if } P \rightarrow P' \text{ and } Q' \rightarrow Q
\]
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>$x &gt; 0$</td>
<td>$x &gt; 1$</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>$x &gt; y$</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></td>
</tr>
<tr>
<td>$x &gt; 0$</td>
<td>$x &gt; 1$</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>$x &gt; y$</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>?</th>
<th>B</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = z$</td>
<td>$\iff$</td>
<td>$z = x$</td>
<td>equivalent</td>
</tr>
<tr>
<td>$x &gt; 0$</td>
<td>$\rightarrow$</td>
<td>$x \geq 0$</td>
<td>$x &gt; 0$</td>
</tr>
<tr>
<td>$x &gt; 0$</td>
<td></td>
<td>$x &gt; 1$</td>
<td></td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td></td>
<td>$x \geq y + 1$</td>
<td></td>
</tr>
<tr>
<td>$p \land q$</td>
<td></td>
<td>$p$</td>
<td></td>
</tr>
<tr>
<td>$p \lor q$</td>
<td></td>
<td>$q$</td>
<td></td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td></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>$x = z \leftrightarrow z = x$</td>
<td>equivalent</td>
<td></td>
</tr>
<tr>
<td>$x &gt; 0 \rightarrow x \geq 0$</td>
<td>$x &gt; 0$</td>
<td></td>
</tr>
<tr>
<td>$x &gt; 0 \leftarrow x &gt; 1$</td>
<td>$x &gt; 1$</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>$x &gt; y$</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>↔</th>
<th>B</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = z$</td>
<td>$\leftrightarrow$</td>
<td>$z = x$</td>
<td>equivalent</td>
</tr>
<tr>
<td>$x &gt; 0$</td>
<td>$\rightarrow$</td>
<td>$x \geq 0$</td>
<td>$x &gt; 0$</td>
</tr>
<tr>
<td>$x &gt; 0$</td>
<td>$\leftarrow$</td>
<td>$x &gt; 1$</td>
<td>$x &gt; 1$</td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td>$\leftrightarrow$</td>
<td>$x \geq y + 1$</td>
<td>equivalent</td>
</tr>
<tr>
<td>$p \land q$</td>
<td></td>
<td>$p$</td>
<td></td>
</tr>
<tr>
<td>$p \lor q$</td>
<td></td>
<td>$q$</td>
<td></td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td></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>↔</th>
<th>B</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x = z \leftrightarrow z = x )</td>
<td></td>
<td>( x \geq 0 \leftrightarrow x \leq 0 )</td>
<td>equivalent</td>
</tr>
<tr>
<td>( x &gt; 0 \rightarrow x \geq 0 )</td>
<td></td>
<td>( x &gt; 0 \leftarrow x &gt; 1 )</td>
<td>( x &gt; 0 )</td>
</tr>
<tr>
<td>( x &gt; y \leftrightarrow x \geq y + 1 )</td>
<td></td>
<td>( p \land q \leftrightarrow p \geq q + 1 )</td>
<td>equivalent</td>
</tr>
<tr>
<td>( p \land q \rightarrow p )</td>
<td></td>
<td>( p \lor q \leftrightarrow p \lor q )</td>
<td>( p ^ \land q )</td>
</tr>
<tr>
<td>( x &gt; y \leftrightarrow x \leq y )</td>
<td></td>
<td>( x \leq y \leftrightarrow x \geq y )</td>
<td>equivalent</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>$x = z \leftrightarrow z = x$</td>
<td>equivalent</td>
<td></td>
</tr>
<tr>
<td>$x &gt; 0 \rightarrow x \geq 0$</td>
<td>$x &gt; 0$</td>
<td></td>
</tr>
<tr>
<td>$x &gt; 0 \leftarrow x &gt; 1$</td>
<td>$x &gt; 1$</td>
<td></td>
</tr>
<tr>
<td>$x &gt; y \leftrightarrow x \geq y + 1$</td>
<td>equivalent</td>
<td></td>
</tr>
<tr>
<td>$p \land q \rightarrow p$</td>
<td>$p \land q$</td>
<td></td>
</tr>
<tr>
<td>$p \lor q \leftarrow q$</td>
<td>$q$</td>
<td></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>(\leftrightarrow)</th>
<th>B</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x = z)</td>
<td>(\leftrightarrow)</td>
<td>(z = x)</td>
<td>equivalent</td>
</tr>
<tr>
<td>(x &gt; 0)</td>
<td>(\rightarrow)</td>
<td>(x \geq 0)</td>
<td>(x &gt; 0)</td>
</tr>
<tr>
<td>(x &gt; 0)</td>
<td>(\leftarrow)</td>
<td>(x &gt; 1)</td>
<td>(x &gt; 1)</td>
</tr>
<tr>
<td>(x &gt; y)</td>
<td>(\leftrightarrow)</td>
<td>(x \geq y + 1)</td>
<td>equivalent</td>
</tr>
<tr>
<td>(p \land q)</td>
<td>(\rightarrow)</td>
<td>(p)</td>
<td>(p \land q)</td>
</tr>
<tr>
<td>(p \lor q)</td>
<td>(\leftarrow)</td>
<td>(q)</td>
<td>(q)</td>
</tr>
<tr>
<td>(x &gt; y)</td>
<td>(\leftarrow)</td>
<td>(x \leq y)</td>
<td>inconsistent</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>?</th>
<th>B</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = z$</td>
<td>$\iff$</td>
<td>$z = x$</td>
<td>equivalent</td>
</tr>
<tr>
<td>$x &gt; 0$</td>
<td>$\rightarrow$</td>
<td>$x \geq 0$</td>
<td>$x &gt; 0$</td>
</tr>
<tr>
<td>$x &gt; 0$</td>
<td>$\leftarrow$</td>
<td>$x &gt; 1$</td>
<td>$x &gt; 1$</td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td>$\iff$</td>
<td>$x \geq y + 1$</td>
<td>equivalent</td>
</tr>
<tr>
<td>$p \land q$</td>
<td>$\rightarrow$</td>
<td>$p$</td>
<td>$p \land q$</td>
</tr>
<tr>
<td>$p \lor q$</td>
<td>$\leftarrow$</td>
<td>$q$</td>
<td>$q$</td>
</tr>
<tr>
<td>$x &gt; y$</td>
<td></td>
<td>$x \leq y$</td>
<td>inconsistent</td>
</tr>
</tbody>
</table>

The stronger relation is a partial order.
Examples Again

\[
\begin{align*}
\{ x &= a \land y = b \} &\quad z &:= x; z := z + y &\{ z &= a + b \} \\
\{ x > y \} &\quad x := 0; y := 0 &\{ x &= y \} \\
\{ x + y \geq 0 \land x > 0 \} &\quad x := x - 1; y := y + 1 &\{ x + y \geq 0 \land x \geq 0 \}
\end{align*}
\]
A simple program with a loop

Property: Given values in variables $x, y \in \mathbb{N}$ such that $y \neq 0$ set $d$ and $r$ such that $x = d \times y + r$ and $r < y$

Specification:

$$\{ x \geq 0 \land y > 0 \} \ S \{ x = d \times y + r \land 0 \leq r < y \}$$
An Informal Argument

Let us now consider the division program

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

Is it correct for \{ x \geq 0 \land y > 0 \} \ S \ { x = d \times y + r \land 0 \leq r < y } \\

For every \( s_{\text{start}} \) that satisfies \( x \geq 0 \land y > 0 \) do we reach a state \( s_{\text{end}} \) that satisfies \( x = d \times y + r \land 0 \leq r < y \) ?

Given any state \( s_{\text{start}} \) what can we say about \( s_{\text{end}} \)?

We cannot say statically how many times the loop will execute!
Recall that

\[ [\text{while } b \text{ do } S] \equiv [\text{if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip}] \]

This means that

\[
\{ P \} \text{ while } b \text{ do } S \{ Q \} \\
\Leftrightarrow \\
\{ P \} \text{ if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip } \{ Q \} \\
\Leftrightarrow \\
\{ P \} \text{ if } b \text{ then } (S; \text{if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip}) \text{ else skip } \{ Q \}
\]

The \( Q \) from the end of one iteration of the loop needs to be the \( P \) for the start of the next iteration.
Recall that

\[ [\text{while } b \text{ do } S] \equiv [\text{if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip}] \]

This means that

\[
\{ P \land b \} \ S \ \{ Q \} \equiv \\
\{ P \land b \} \ S; \text{while } b \text{ do } S \ \{ Q \} \equiv \\
\{ P \land b \} \ S; \text{if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip } \ \{ Q \}
\]

The \( Q \) from the end of one iteration of the loop needs to be the \( P \) for the start of the next iteration.
Recall that

\[ [\text{while } b \text{ do } S] \equiv [\text{if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip}] \]

This means that

\[
\{ P \land b \} S \{ Q \} \iff \{ P \land b \} S \{ P \land b \} \text{ while } b \text{ do } S \{ Q \}
\]

The \( Q \) from the end of one iteration of the loop needs to be the \( P \) for the start of the next iteration.
Recall that

\[ [\text{while } b \text{ do } S] \equiv [\text{if } b \text{ then }(S;\text{while } b \text{ do } S) \text{ else skip}] \]

This means that

\[
\{ P \} \text{ while } b \text{ do } S \{ P \land \neg b \} \\
\Leftrightarrow \\
\{ P \} \text{ if } b \text{ then } (S;\text{while } b \text{ do } S) \text{ else skip} \{ P \land \neg b \} \\
\Leftrightarrow \\
\{P\text{if } b \text{ then } (S;\text{if } b \text{ then } (S;\text{while } b \text{ do } S) \text{ else skip}) \text{ else skip}\} P \land \neg b
\]

\( P \) is an expression that is true on every iteration of the loop; it is a loop invariant.
Let us now consider the division program

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

What is a loop invariant for this loop?

- The variables \(d\) and \(r\) are changed so we need to mention them.
- We increase \(d\) by 1 and decrease \(r\) by \(y\).
- So \(d \times y\) increases as much as \(r\) decreases.
- So \(K = d \times y + r\) is invariant, but what is \(K\)?
- It needs to hold at the start, where \(r = x\) and \(d = 0\).
- So \(K = 0 \times y + x = x\).
- Making our invariant \(x = (d \times y) + r\), which is unsurprising.
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 \textit{loop invariant}.

We motivated this informally before. Now let’s see it in action on the division program!