# Programm- & Systemverifikation

## Hoare Logic

**Georg Weissenbacher**

**184.741**

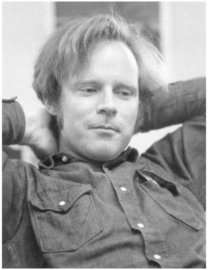
# What happened so far

- How bugs come into being:
  - Fault – cause of an error (e.g., mistake in coding)
  - Error – *incorrect* state that may lead to failure
  - Failure – deviation from *desired* behaviour
- We specified *intended* behaviour using assertions
- We proved our programs correct (inductive invariants).
- We learned how to test programs.
- We heard about logical formalisms:
  - Propositional Logic
  - First Order Logic

## Instructions and Assertions

- FOL and derivations enable us to reason about assertions
- But what about *instructions*?
  - Currently, we can't refer to instructions in inference rules

# Assigning Meaning to Programs [Floyd67]



Robert W. Floyd
(1936-2001)

- Used assertions to attach logical interpretations to programs
- proof "by induction on the number of commands executed"

# Assigning Meaning to Programs [Floyd67]

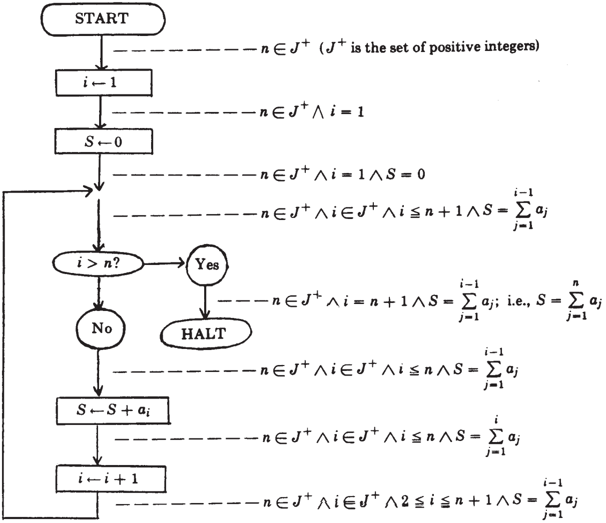
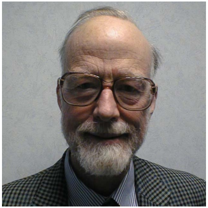



FIGURE 1. Flowchart of program to compute $S = \sum_{j=1}^{n} a_j$ ($n \geqq 0$)

## An Axiomatic Basis for Computer Programming [Hoare69]



Sir Tony Hoare
(1934–)

- “Hoare Logic” (aka Floyd-Hoare Logic)
- System of *axioms* and inference rules for program verification

# Hoare Triples

$$\{P\}\ C\ \{Q\}$$

## Definition (Hoare Triple)

A Hoare triple comprises a pre-condition, a statement, and a post-condition.

The Hoare Triple

$$\{P\}\ C\ \{Q\}$$

means that if $C$ is executed in a state for which $P$ holds, then $Q$ is true for any state in which $C$ may halt.

We refer to $P$ as the *pre-condition* and to $Q$ as the *post-condition* of the Hoare Triple.

# Hoare Triples

$$\{P\}\ C\ \{Q\}$$

As an FOL instance (schematic):

$$\forall x_0, y_0, \ldots, x_1, y_1, \ldots .$$
$$P(x_0, y_0, \ldots) \wedge C(x_0, y_0, \ldots, x_1, y_1, \ldots) \rightarrow Q(x_1, y_1, \ldots)$$

A Hoare triple characterizes the effect of commands on assertions

# Hoare's Axioms: Skip statement

$$\frac{}{\{P\}\ \texttt{skip}\ \{P\}}$$

Example:

- $\{x > 10\}\ \texttt{skip}\ \{x > 10\}$
- one example should really be enough ;-)

# Hoare's Axioms: Assignment

$$\frac{}{\{Q[E/x]\}\ x{:=}E\ \{Q\}}$$

Intuition:

- $Q$ holds for new value of $x$
- $E$ evaluated in old state determines new value of $x$
- Therefore, $Q[E/x]$ must hold before execution

# Hoare's Axioms: Assignment

$$\overline{\{Q[E/x]\}\ x{:=}E\ \{Q\}}$$

Examples:

- $\{x > 10\}\ y{:=}10\ \{x > y\}$
- $\{x > y + 1\}\ y{:=}y + 1\ \{x > y\}$
- $\{\exists z\,.\,z > x + 1\}\ y{:=}x + 1\ \{\exists x\,.\,x > y\}$
    - Rename quantified $x$ to avoid clash!

# Hoare's Axioms: Assignment

Remember from our lecture on assertions:

```
y = y-1;

x = x-1;
assert (x>y);
```

# Hoare's Axioms: Assignment

Remember from our lecture on assertions:

```
y = y-1;
assert (x-1>y);
x = x-1;
assert (x>y);
```

## Hoare's Axioms: Assignment

Remember from our lecture on assertions:

```
assert (x-1>y-1);
y = y-1;
assert (x-1>y);
x = x-1;
assert (x>y);
```

# Hoare's Axioms: Composition

$$\frac{\{P\}\ C_1\ \{R\} \qquad \{R\}\ C_2\ \{Q\}}{\{P\}\ C_1; C_2\ \{Q\}}$$

Example:

$$\frac{\{(x+1) \leq 10\}\ x := x+1\ \{x \leq 10\}\ \{x \leq 10\}\ y := 0\ \{x \leq 10\}}{\{(x+1) \leq 10\}\ x := x+1;\ y := 0\ \{x \leq 10\}}$$

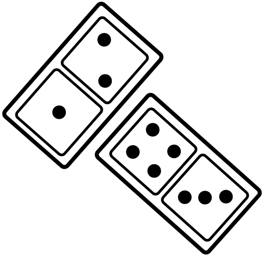
# Hoare's Axioms: Composition

$$\frac{\{P\}\ C_1\ \{R\} \qquad \{R\}\ C_2\ \{Q\}}{\{P\}\ C_1; C_2\ \{Q\}}$$

Example:

$$\frac{\{(x+1) \leq 10\}\ x := x+1\ \{x \leq 10\}\ \{x \leq 10\}\ y := 0\ \{x \leq 10\}}{\{(x+1) \leq 10\}\ x := x+1;\ y := 0\ \{x \leq 10\}}$$

Note: "intermediate" assertion must be *exact match*!



# Hoare's Axioms: Composition

We have *implicitly* applied this before:

```
assert (x-1>y-1);
y = y-1;
assert (x-1>y);
```

```
assert (x-1>y);
x = x-1;
assert (x>y);
```

# Hoare's Axioms: Composition

We have *implicitly* applied this before:

```
assert (x-1>y-1);
y = y-1;
```

```
x = x-1;
assert (x>y);
```

Enables us to *eliminate* intermediate assertions!

# Hoare's Axioms: Conditional

$$\frac{\{B \wedge P\}\ C_1\ \{Q\} \qquad \{\neg B \wedge P\}\ C_2\ \{Q\}}{\{P\}\ \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\ \{Q\}}$$

Example:

$$\frac{\{\mathrm{even}(x)\}\ x := x + 1\ \{\mathrm{odd}(x)\} \qquad \{\neg\,\mathrm{even}(x)\}\ \texttt{skip}\ \{\mathrm{odd}(x)\}}{\{\mathrm{true}\}\ \texttt{if}\ (\mathrm{even}(x))\ \texttt{then}\ x := x + 1\ \texttt{else}\ \ \texttt{skip}\ \{\mathrm{odd}(x)\}}$$

# Hoare's Axioms: Conditional

$$\frac{\{B \wedge P\}\ C_1\ \{Q\} \qquad \{\neg B \wedge P\}\ C_2\ \{Q\}}{\{P\}\ \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\ \{Q\}}$$

Example:

$$\frac{\{\mathrm{even}(x)\}\ x := x + 1\ \{\mathrm{odd}(x)\} \qquad \{\neg\,\mathrm{even}(x)\}\ \texttt{skip}\ \{\mathrm{odd}(x)\}}{\{\mathrm{true}\}\ \texttt{if}\ (\mathrm{even}(x))\ \texttt{then}\ x := x + 1\ \texttt{else}\ \texttt{skip}\ \{\mathrm{odd}(x)\}}$$

- Where (or what) is $P$ in this example?

# Hoare's Axioms: Conditional

$$\frac{\{B \wedge P\}\ C_1\ \{Q\} \qquad \{\neg B \wedge P\}\ C_2\ \{Q\}}{\{P\}\ \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\ \{Q\}}$$

Example:

```
if (n % 2) then
  z = z + y;
else
  skip;

y = y * 2;

n = n / 2;
```

{ m * x = z + (n * y) }

# Hoare's Axioms: Conditional

$$\frac{\{B \wedge P\}\ C_1\ \{Q\} \qquad \{\neg B \wedge P\}\ C_2\ \{Q\}}{\{P\}\ \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\ \{Q\}}$$

Example:

```
if (n % 2) then
  z = z + y;
else
  skip;

y = y * 2;
```

$\{\ m * x = z + (\lfloor \frac{n}{2} \rfloor * y)\ \}$

```
n = n / 2;
```

$\{\ m * x = z + (n * y)\ \}$

# Hoare's Axioms: Conditional

$$\frac{\{B \wedge P\}\ C_1\ \{Q\} \qquad \{\neg B \wedge P\}\ C_2\ \{Q\}}{\{P\}\ \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\ \{Q\}}$$

Example:

```
if (n % 2) then
  z = z + y;
else
  skip;
```

$\{\ m * x = z + (\lfloor \frac{n}{2} \rfloor * 2 * y)\ \}$

```
y = y * 2;
```

$\{\ m * x = z + (\lfloor \frac{n}{2} \rfloor * y)\ \}$

```
n = n / 2;
```

$\{\ m * x = z + (n * y)\ \}$

# Hoare's Axioms: Conditional

$$\frac{\{B \wedge P\}\ C_1\ \{Q\} \qquad \{\neg B \wedge P\}\ C_2\ \{Q\}}{\{P\}\ \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\ \{Q\}}$$

Example:

```
if (n % 2) then
  z = z + y;
else
  skip;
```

$\{\ m * x = z + (\lfloor \frac{n}{2} \rfloor * 2 * y)\ \}$

```
y = y * 2;
```

$\{\ m * x = z + (\lfloor \frac{n}{2} \rfloor * y)\ \}$

```
n = n / 2;
```

$\{\ m * x = z + (n * y)\ \}$

# Hoare's Axioms: Conditional

$$\frac{\{B \wedge P\}\ C_1\ \{Q\} \qquad \{\neg B \wedge P\}\ C_2\ \{Q\}}{\{P\}\ \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\ \{Q\}}$$

Example:

```
if (n % 2) then
  z = z + y;
else
  skip;
```

# Hoare's Axioms: Conditional

$$\frac{\{B \wedge P\}\ C_1\ \{Q\} \qquad \{\neg B \wedge P\}\ C_2\ \{Q\}}{\{P\}\ \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\ \{Q\}}$$

Example:

```
if (n % 2) then
  z = z + y;
else
  skip;
```

Need to prove:

$$\{(n\%2 = 1) \wedge (m * x = z + (n * y))\}$$
$$\mathtt{z} = \mathtt{z} + \mathtt{y}$$
$$\{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

# Hoare's Axioms: Conditional (continued)

$$\{(n\%2 = 1) \wedge (m * x = z + (n * y))\}$$
$$\mathtt{z = z + y}$$
$$\{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

Rewrite pre-condition:

$$(n\%2 = 1) \wedge (m * x = z + (n * y))$$

# Hoare's Axioms: Conditional (continued)

$$\{(n\%2 = 1) \wedge (m * x = z + (n * y))\}$$
$$\mathtt{z = z + y}$$
$$\{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

Rewrite pre-condition:

$$\begin{aligned} & (n\%2 = 1) \wedge (m * x = z + (n * y)) \\ \equiv\ & (n\%2 = 1) \wedge (m * x = z + y + ((n - 1) * y)) \end{aligned}$$

# Hoare's Axioms: Conditional (continued)

$$\{(n\%2 = 1) \wedge (m * x = z + (n * y))\}$$
$$\mathtt{z = z + y}$$
$$\{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

Rewrite pre-condition:

$$\begin{aligned} & (n\%2 = 1) \wedge (m * x = z + (n * y)) \\ \equiv\ & (n\%2 = 1) \wedge (m * x = z + y + ((n - 1) * y)) \\ \equiv\ & (n\%2 = 1) \wedge \left(m * x = z + y + (\frac{n - 1}{2} * 2 * y)\right) \end{aligned}$$

# Hoare's Axioms: Conditional (continued)

$$\{(n\%2 = 1) \wedge (m * x = z + (n * y))\}$$
$$\mathtt{z = z + y}$$
$$\{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

Rewrite pre-condition:

$$\begin{aligned} & (n\%2 = 1) \wedge (m * x = z + (n * y)) \\ \equiv\ & (n\%2 = 1) \wedge (m * x = z + y + ((n - 1) * y)) \\ \equiv\ & (n\%2 = 1) \wedge \left(m * x = z + y + (\frac{n-1}{2} * 2 * y)\right) \\ \equiv\ & (n\%2 = 1) \wedge \left(m * x = z + y + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\right) \end{aligned}$$

# Hoare's Axioms: Conditional (continued)

$$\{(n\%2 = 1) \wedge (m * x = z + (n * y))\}$$
$$\mathrm{z = z + y}$$
$$\{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

Rewrite pre-condition:

$$\begin{aligned} & (n\%2 = 1) \wedge (m * x = z + (n * y)) \\ \equiv\ & (n\%2 = 1) \wedge (m * x = z + y + ((n-1) * y)) \\ \equiv\ & (n\%2 = 1) \wedge \left(m * x = z + y + (\frac{n-1}{2} * 2 * y)\right) \\ \equiv\ & (n\%2 = 1) \wedge \left(m * x = z + y + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\right) \end{aligned}$$

Note: Using the assignment rule, we can prove the following:

$$\{m * x = z + y + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$
$$\mathrm{z = z + y}$$
$$\{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

# Hoare's Axioms: Conditional (continued)

But the pre-condition of

$$\{(n\%2 = 1) \wedge \left(m * x = z + y + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\right)\}$$

$$\mathtt{z = z + y}$$

$$\{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

is *stronger* than the precondition of

$$\{m * x = z + y + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

$$\mathtt{z = z + y}$$

$$\{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

# Hoare's Axioms: Consequence

$$\frac{P' \to P \quad \{P\}\ S\ \{Q\} \quad Q \to Q'}{\{P'\}\ S\ \{Q'\}}$$

It is legal to

- strengthen pre-condition, and
- weaken the post-condition

Also allows us to combine Hoare Logic and FOL derivations:

$$\frac{\{\text{even}(x)\}\ x := x + 1\ \{\neg\,\text{even}(x)\} \qquad \dfrac{\neg\,\text{even}(x) \vdash \text{odd}(x)}{\neg\,\text{even}(x) \to \text{odd}(x)}}{\{\text{even}(x)\}\ x := x + 1\ \{\text{odd}(x)\}}$$

and

$$\frac{\neg\,\text{even}(x) \qquad \dfrac{\forall n\,.\ \text{even}(n) \lor \text{odd}(n)}{\text{even}(x) \lor \text{odd}(x)}}{\text{odd}(x)}$$

# Hoare's Axioms: Previous Example Revisited

We are allowed to *strengthen* the the pre-condition:

$$\{ \qquad\qquad\qquad \left(m * x = z + y + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\right)\}$$

$$\mathrm{z} = \mathrm{z} + \mathrm{y}$$

$$\{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

## Hoare's Axioms: Previous Example Revisited

We are allowed to *strengthen* the the pre-condition:

$$\{\,(n\%2 = 1) \wedge \left(m * x = z + y + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\right)\}$$

$$\mathrm{z} = \mathrm{z} + \mathrm{y}$$

$$\{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

## Hoare's Axioms: Previous Example Revisited

We are allowed to *strengthen* the the pre-condition:

$$\{\, (n\%2 = 1) \wedge \left(m * x = z + y + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\right)\}$$

$$\mathrm{z = z + y}$$

$$\{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

Since

$$(n\%2 = 1) \wedge \left(m * x = z + y + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\right) \equiv$$

$$(n\%2 = 1) \wedge (m * x = z + (n * y))$$

we obtain

$$\{(n\%2 = 1) \wedge \left(m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\right)\}$$

$$\mathrm{z = z + y}$$

$$\{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

# Hoare's Axioms: Previous Example Revisited

Similarly, we strengthen pre-condition of else-branch:

$$\{ \qquad \Big(m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\Big)\} \ \texttt{skip} \ \{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

# Hoare's Axioms: Previous Example Revisited

Similarly, we strengthen pre-condition of else-branch:

$$\{ (n\%2 = 0) \wedge \left(m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\right)\} \ \texttt{skip} \ \{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

# Hoare's Axioms: Previous Example Revisited

Similarly, we strengthen pre-condition of else-branch:

$$\{\,(n\%2=0)\wedge\left(m*x=z+(\frac{n}{2}*2*y)\right)\}\ \texttt{skip}\ \{m*x=z+(\left\lfloor\frac{n}{2}\right\rfloor*2*y)\}$$

## Hoare's Axioms: Previous Example Revisited

Similarly, we strengthen pre-condition of else-branch:

$$\{\,(n\%2 = 0) \land \,(m * x = z + (n * y))\}\ \texttt{skip}\ \{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}$$

# Hoare's Axioms: Previous Example Revisited

Similarly, we strengthen pre-condition of else-branch:

$$\{\,(n\%2 = 0) \wedge \,(m * x = z + (n * y))\}\ \texttt{skip}\ \{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}\ \checkmark$$

# Hoare's Axioms: Example for Conditional Rule

We can now apply

$$\frac{\{B \wedge P\}\ C_1\ \{Q\} \qquad \{\neg B \wedge P\}\ C_2\ \{Q\}}{\{P\}\ \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\ \{Q\}}$$

to $C \quad \stackrel{\text{def}}{=} \quad \texttt{if}\ (\mathtt{n}\%2)\ \texttt{then}\ \mathtt{z} = \mathtt{z} + \mathtt{y};\ \texttt{else}\ \texttt{skip};$

$$\frac{\begin{array}{ll} \{(n\%2 = 0) \wedge (m * x = z + (n * y))\} & \{(n\%2 = 1) \wedge (m * x = z + (n * y))\} \\ \texttt{skip} & \mathtt{z} = \mathtt{z} + \mathtt{y} \\ \{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\} & \{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\} \end{array}}{\{m * x = z + (n * y)\}\ C\ \{m * x = z + (\left\lfloor \frac{n}{2} \right\rfloor * 2 * y)\}}$$

# Hoare's Axioms: Example Revisited

Finally, we obtain:

$\{m * x = z + (n * y)\}$

```
if (n % 2) then
  z = z + y;
else
  skip;
```

$\{ m * x = z + (\lfloor \frac{n}{2} \rfloor * 2 * y) \}$

$\{ m * x = z + (\lfloor \frac{n}{2} \rfloor * 2 * y) \}$

```
y = y * 2;
n = n / 2;
```

$\{ m * x = z + (n * y) \}$

## Hoare's Axioms: Example Revisited

Finally, we obtain:

{m * x = z + (n * y)}

```
if (n % 2) then
  z = z + y;
else
  skip;
 y = y * 2;
n = n / 2;
```

{ m * x = z + (n * y) }

Therefore, $m * x = z + (n * y)$ is an invariant for the loop body!

# Hoare's Axioms: Example Revisited

Finally, we obtain:

```
{m * x = z + (n * y)}
if (n % 2) then
  z = z + y;
else
  skip;
 y = y * 2;
n = n / 2;
{ m * x = z + (n * y) }
```

Therefore, $m * x = z + (n * y)$ is an invariant for the loop body!

- But we don't have a rule for loops yet!

# While Loops

$$\frac{\{P \wedge B\}\ C\ \{P\}}{\{P\}\ \text{while}\ B\ \text{do}\ C\ \{\neg B \wedge P\}}$$

- Statement $S$ doesn't change $P$ ($P$ is *invariant*)
- $P$ holds upon loop entry and exit (*loop invariant*)

Example:

$$\frac{\{(x \neq 0) \wedge (x \geq 0)\}\ x := x - 1\ \{x \geq 0\}}{\{x \geq 0\}\ \texttt{while}\ (x \neq 0)\ \text{do}\ x := x - 1\ \{\neg(x \neq 0) \wedge (x \geq 0)\}}$$

# While Loops

In context of a larger proof:

$$\dfrac{\dfrac{\dfrac{(x \neq 0) \wedge (x \geq 0) \vdash (x - 1 \geq 0)}{(x \neq 0) \wedge (x \geq 0) \rightarrow (x - 1 \geq 0)} \qquad \{(x - 1 \geq 0)\}\ x := x - 1\ \{x \geq 0\}}{\{(x \neq 0) \wedge (x \geq 0)\}\ x := x - 1\ \{x \geq 0\}}}{\{x \geq 0\}\ \texttt{while}\ (x \neq 0)\ \text{do}\ x := x - 1\ \{\neg(x \neq 0) \wedge (x \geq 0)\}}$$

Here, we derive

- $\{(x - 1 \geq 0)\}\ x := x - 1\ \{x \geq 0\}$ using Hoare's assignment rule
- $\{(x \neq 0) \wedge (x \geq 0)\}\ x := x - 1\ \{x \geq 0\}$ using rule of consequence
- $\{x \geq 0\}\ \texttt{while}\ (x \neq 0)\ \text{do}\ x := x - 1\ \{\neg(x \neq 0) \wedge (x \geq 0)\}$ using loop rule

## While Loops

In context of a larger proof:

$$\dfrac{\dfrac{\dfrac{(x \neq 0) \wedge (x \geq 0) \vdash (x - 1 \geq 0)}{(x \neq 0) \wedge (x \geq 0) \rightarrow (x - 1 \geq 0)} \qquad \{(x - 1 \geq 0)\}\ x := x - 1\ \{x \geq 0\}}{\{(x \neq 0) \wedge (x \geq 0)\}\ x := x - 1\ \{x \geq 0\}}}{\{x \geq 0\}\ \texttt{while}\ (x \neq 0)\ \text{do}\ x := x - 1\ \{\neg(x \neq 0) \wedge (x \geq 0)\}}$$

Here, we derive

- $\{(x - 1 \geq 0)\}\ x := x - 1\ \{x \geq 0\}$ using Hoare's assignment rule
- $\{(x \neq 0) \wedge (x \geq 0)\}\ x := x - 1\ \{x \geq 0\}$ using rule of consequence
- $\{x \geq 0\}\ \texttt{while}\ (x \neq 0)\ \text{do}\ x := x - 1\ \{\neg(x \neq 0) \wedge (x \geq 0)\}$ using loop rule

Don't forget: we still have to discharge the assumption

$$(x \neq 0) \wedge (x \geq 0) \vdash (x - 1 \geq 0)$$

to show that the Hoare triple is valid. Requires the theory of integer arithmetic (not presented here).

# Hoare Rules: Example Revisited

We can now apply this to the loop body from before:

```
while (n ≠ 0)
  {m * x = z + (n * y)}
  if (n % 2) then
    z = z + y;
  else
    skip;
  y = y * 2;
  n = n / 2;
  { m * x = z + (n * y) }
```

## Hoare Rules: Example Revisited

We can now apply this to the loop body from before:

{m * x = z + (n * y)}

```
while (n ≠ 0)
```

  {m * x = z + (n * y)}

```
  if (n % 2) then
    z = z + y;
  else
    skip;
  y = y * 2;
  n = n / 2;
```

  { m * x = z + (n * y) }

$\{(n = 0) \wedge (m * x = z + (n * y))\}$

## Hoare Rules: Example Revisited

We can now apply this to the loop body from before:

```
{m * x = z + (n * y)}
while (n ≠ 0)
  {m * x = z + (n * y)}
  if (n % 2) then
    z = z + y;
  else
    skip;
  y = y * 2;
  n = n / 2;
  { m * x = z + (n * y) }
{(n = 0) ∧ (m * x = z + (n * y))}
```

and weaken the post-condition to $(z = m * x)$.

# Hoare Rules: Overview

$$\frac{}{\{P[E/\mathtt{x}]\}\ \mathtt{x:=}E\ \{P\}}$$

$$\frac{\{P\}\ C_1\ \{Q\}\ ,\ \{Q\}\ C_2\ \{R\}}{\{P\}\ C_1\ ;\ C_2\ \{R\}}$$

$$\frac{\{B \wedge P\}\ C_1\ \{Q\}\quad \{\neg B \wedge P\}\ C_2\ \{Q\}}{\{P\}\ \text{if } B \text{ then } C_1 \text{ else } C_2\ \{Q\}}$$

$$\frac{P' \rightarrow P \quad \{P\}\ C\ \{Q\} \quad Q \rightarrow Q'}{\{P'\}\ C\ \{Q'\}}$$

$$\frac{\{P \wedge B\}\ C\ \{P\}}{\{P\}\ \text{while } B \text{ do } C\ \{\neg B \wedge P\}}$$

## Example: Finding an Inductive Invariant

- Add an *inductive invariant* to the code
- Use it to show that the assertion after the loop holds
- Add comments to the code explaining
  - why your assertion is an inductive invariant
  - why it shows that the assertion after the loop holds

```
int x = i;
int y = j;
while (x != 0)
{
  x--;
  y--;
  assert (?);
}
assert ((i != j) || (y == 0));
```

# Example: Finding an Inductive Invariant

We want to prove:

$\{\text{true}\}$

```
x = i;
y = j;
while (x != 0)
  x = x - 1;
  y = y - 1;
```

$\{(i \neq j) \vee (y = 0)\}$

One possible invariant is $(i - j) = (x - y)$.

## Example: Finding an Inductive Invariant

We want to prove:

{true}

```
x = i;
y = j;
while (x != 0)
  x = x - 1;
  y = y - 1;
```

$\{(i \neq j) \vee (y = 0)\}$

## Example: Finding an Inductive Invariant

We want to prove:

$\{\text{true}\}$

```
x = i;

y = j;

while (x != 0)

  x = x - 1;

  y = y - 1;
```

$\{(i - j) = (x - y)\}$

$\{(i \neq j) \vee (y = 0)\}$

## Example: Finding an Inductive Invariant

We want to prove:

$\{\text{true}\}$

```
x = i;

y = j;

while (x != 0)

  x = x - 1;
```

$\{(i - j) = (x - (y - 1))\}$

```
  y = y - 1;
```

$\{(i - j) = (x - y)\}$

$\{(i \neq j) \vee (y = 0))\}$

# Example: Finding an Inductive Invariant

We want to prove:

$\{\text{true}\}$

```
x = i;

y = j;

while (x != 0)
```

$\{(i - j) = ((x - 1) - (y - 1))\}$

```
  x = x - 1;
```

$\{(i - j) = (x - (y - 1))\}$

```
  y = y - 1;
```

$\{(i - j) = (x - y)\}$

$\{(i \neq j) \vee (y = 0))\}$

## Example: Finding an Inductive Invariant

We want to prove:

$\{\text{true}\}$

```
x = i;

y = j;
```

$\{(i - j) = (x - y)\}$

```
while (x != 0)
```

$\{(i - j) = (x - y)\}$

```
  x = x - 1;

  y = y - 1;
```

$\{(i - j) = (x - y)\}$

$\{(x = 0) \wedge ((i - j) = (x - y))\}$

$\{(i \neq j) \vee (y = 0))\}$

## Example: Finding an Inductive Invariant

We want to prove:

$\{\text{true}\}$

```
x = i;
```
$\{(i - j) = (x - j)\}$
```
y = j;
```
$\{(i - j) = (x - y)\}$
```
while (x != 0)
```
  $\{(i - j) = (x - y)\}$
```
  x = x - 1;

  y = y - 1;
```
  $\{(i - j) = (x - y)\}$

$\{(x = 0) \wedge ((i - j) = (x - y))\}$

$\{(i \neq j) \vee (y = 0))\}$

# Example: Finding an Inductive Invariant

We want to prove:

$\{\text{true}\}$
$\{(i - j) = (i - j)\}$
```
x = i;
```
$\{(i - j) = (x - j)\}$
```
y = j;
```
$\{(i - j) = (x - y)\}$
```
while (x != 0)
```
  $\{(i - j) = (x - y)\}$
```
  x = x - 1;

  y = y - 1;
```
  $\{(i - j) = (x - y)\}$
$\{(x = 0) \wedge ((i - j) = (x - y))\}$
$\{(i \neq j) \vee (y = 0))\}$

## Example: Finding an Inductive Invariant

We want to prove:

$\{\text{true}\}$
$\{\text{true}\}$
```
x = i;
```
$\{(i - j) = (x - j)\}$
```
y = j;
```
$\{(i - j) = (x - y)\}$
```
while (x != 0)
```
  $\{(i - j) = (x - y)\}$
```
  x = x - 1;

  y = y - 1;
```
  $\{(i - j) = (x - y)\}$

$\{(x = 0) \wedge ((i - j) = (x - y))\}$
$\{(i \neq j) \vee (y = 0))\}$

## Example: Finding an Inductive Invariant

We want to prove:

$\{\text{true}\}$

$\{\text{true}\}$

```
x = i;
```

$\{(i - j) = (x - j)\}$

```
y = j;
```

$\{(i - j) = (x - y)\}$

```
while (x != 0)
```

$\{(i - j) = (x - y)\}$

```
  x = x - 1;

  y = y - 1;
```

$\{(i - j) = (x - y)\}$

$\{(x = 0) \wedge ((i - j) = (0 - y))\}$

$\{(i \neq j) \vee (y = 0))\}$

## Want something more challenging?

If you know the invariant, this was rather easy…

- (easy enough for an exam)

# Want something more challenging?

If you know the invariant, this was rather easy…

- (easy enough for an exam)

Let's try something harder!

## Greatest Common Divisor

Let

$$\forall x, y, z \,.\, \text{divides}(x, y, z) \Leftrightarrow ((x \% z = 0) \wedge (y \% z = 0))$$

$$\forall x, y, z \,.\, (z = GCD(x, y)) \Leftrightarrow \text{divides}(x, y, z) \wedge$$
$$(\nexists r \,.\, (r > z) \wedge \text{divides}(x, y, r))$$

- (assuming $\mathcal{D} = \mathbb{N}$ and % is the modulo operator)

## A Hoare Logic Proof of Euclid's Algorithm

We want to show the following:

$\{\text{true}\}$

```
if (x > y) then
  k := x;
  m := y;
else
  k := y;
  m := x;
while (m ≠ 0) do
  r := k%m;
  k := m;
  m := r;
```

$\{GCD(x, y) = k\}$

# A Hoare Logic Proof of Euclid's Algorithm

- Break the proof into sub-proofs
- Use compact "in-line" presentation:

$\{P\}$
$C_1$
$\{Q\}$
$C_2$
$\{R\}$

$$\frac{\{P\}\ C_1\ \{Q\} \qquad \{Q\}\ C_2\ \{R\}}{\{P\}\ C_1 ; C_2\ \{R\}}$$

# Euclid's Algorithm – Loop Invariant

Assume we have a *predicate GCD* with the following axioms

- $\forall x, y \,.\, GCD(x, y) = GCD(y, x)$
- $\forall x \,.\, GCD(0, x) = x$
- $\forall x \,.\, GCD(x, x) = x$
- $\forall x, y \,.\, (x > y) \rightarrow GCD(x, y) = GCD(x \% y, y)$

We will first show the following:

$\{(k \geq m) \land GCD(x, y) = GCD(m, k)\}$

```
while (m != 0) {
  r = k % m;
  k = m;
  m = r;
}
```

$\{GCD(x, y) = k\}$

# Euclid's Algorithm – Loop Invariant

Assume we have a *predicate GCD* with the following axioms

- $\forall x, y \,.\, GCD(x, y) = GCD(y, x)$
- $\forall x \,.\, GCD(0, x) = x$
- $\forall x \,.\, GCD(x, x) = x$
- $\forall x, y \,.\, (x > y) \rightarrow GCD(x, y) = GCD(x \% y, y)$

```
while (m != 0) {

  r = k % m;

  k = m;

  m = r;

}
```

# Euclid's Algorithm – Loop Invariant

Assume we have a *predicate GCD* with the following axioms

- $\forall x, y \,.\, GCD(x, y) = GCD(y, x)$
- $\forall x \,.\, GCD(0, x) = x$
- $\forall x \,.\, GCD(x, x) = x$
- $\forall x, y \,.\, (x > y) \rightarrow GCD(x, y) = GCD(x \% y, y)$

```
while (m != 0) {

  r = k % m;

  k = m;

  m = r;
  {(k ≥ m) ∧ GCD(x, y) = GCD(k, m)}
}
```

## Euclid's Algorithm – Loop Invariant

Assume we have a *predicate GCD* with the following axioms

- $\forall x, y \,.\, GCD(x, y) = GCD(y, x)$
- $\forall x \,.\, GCD(0, x) = x$
- $\forall x \,.\, GCD(x, x) = x$
- $\forall x, y \,.\, (x > y) \rightarrow GCD(x, y) = GCD(x\%y, y)$

```
while (m != 0) {

  r = k % m;

  k = m;
  {(k ≥ r) ∧ GCD(x, y) = GCD(k, r)}
  m = r;
  {(k ≥ m) ∧ GCD(x, y) = GCD(k, m)}
}
```

# Euclid's Algorithm – Loop Invariant

Assume we have a *predicate GCD* with the following axioms

- $\forall x, y \,.\, GCD(x, y) = GCD(y, x)$
- $\forall x \,.\, GCD(0, x) = x$
- $\forall x \,.\, GCD(x, x) = x$
- $\forall x, y \,.\, (x > y) \rightarrow GCD(x, y) = GCD(x\%y, y)$

```
while (m != 0) {

  r = k % m;
  {(m ≥ r) ∧ GCD(x, y) = GCD(m, r)}
  k = m;
  {(k ≥ r) ∧ GCD(x, y) = GCD(k, r)}
  m = r;
  {(k ≥ m) ∧ GCD(x, y) = GCD(k, m)}
}
```

# Euclid's Algorithm – Loop Invariant

Assume we have a *predicate GCD* with the following axioms

- $\forall x, y \,.\, GCD(x, y) = GCD(y, x)$
- $\forall x \,.\, GCD(0, x) = x$
- $\forall x \,.\, GCD(x, x) = x$
- $\forall x, y \,.\, (x > y) \rightarrow GCD(x, y) = GCD(x\%y, y)$

```
while (m != 0) {
  {(m ≥ (k%m)) ∧ GCD(x, y) = GCD(m, (k%m))}
  r = k % m;
  {(m ≥ r) ∧ GCD(x, y) = GCD(m, r)}
  k = m;
  {(k ≥ r) ∧ GCD(x, y) = GCD(k, r)}
  m = r;
  {(k ≥ m) ∧ GCD(x, y) = GCD(k, m)}
}
```

# Euclid's Algorithm – Loop Invariant

Assume we have a *predicate GCD* with the following axioms

- $\forall x, y \,.\, GCD(x, y) = GCD(y, x)$
- $\forall x \,.\, GCD(0, x) = x$
- $\forall x \,.\, GCD(x, x) = x$
- $\forall x, y \,.\, (x > y) \rightarrow GCD(x, y) = GCD(x\%y, y)$

$$\texttt{while (m != 0) } \{$$

$$\{\underbrace{(m \geq (k\%m))}_{\texttt{true}} \wedge GCD(x, y) = GCD(m, (k\%m))\}$$

```
  r = k % m;
```

$$\{(m \geq r) \wedge GCD(x, y) = GCD(m, r)\}$$

```
  k = m;
```

$$\{(k \geq r) \wedge GCD(x, y) = GCD(k, r)\}$$

```
  m = r;
```

$$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$$

$$\}$$

# Euclid's Algorithm – Loop Invariant

Assume we have a *predicate GCD* with the following axioms

- $\forall x, y \,.\, GCD(x, y) = GCD(y, x)$
- $\forall x \,.\, GCD(0, x) = x$
- $\forall x \,.\, GCD(x, x) = x$
- $\forall x, y \,.\, (x > y) \rightarrow GCD(x, y) = GCD(x \% y, y)$

$$
\frac{\begin{array}{c} (k \geq m) \wedge GCD(x, y) = GCD(k, m) \\ \rightarrow \\ (GCD(x, y) = GCD(m, (k\%m)) \end{array} \qquad \begin{array}{c} \{GCD(x, y) = GCD(m, (k\%m))\} \\ r := k\%m;\ k := m;\ m = r \\ \{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\} \end{array}}{\begin{array}{c} \{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\} \\ r := k\%m;\ k := m;\ m = r \\ \{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\} \end{array}}
$$

# Proof by Case Split

We still need to show that

$$((k \geq m) \wedge (GCD(x, y) = GCD(k, m))) \rightarrow (GCD(x, y) = GCD(m, (k\%m))$$

We need the following axioms of the theory of arithmetic:

$$\forall x, y \,.\, (x \geq y) \Leftrightarrow ((x = y) \vee (x > y))$$
$$\forall x, y \,.\, (x\%x) = 0$$
$$\forall x, y, z \,.\, (x = y) \wedge (y = z) \rightarrow (x = z)$$

Perform a *case split.* First for $(k > m)$:

$$\frac{(k > m) \wedge (GCD(x, y) = GCD(m, k)) \qquad (k > m) \rightarrow GCD(k, m) = GCD(k\%m, m)}{(k > m) \wedge (GCD(x, y) = GCD(k\%m, m))}$$

$$\frac{(k > m) \wedge (GCD(x, y) = GCD(k\%m, m)) \quad GCD(k\%m, m) = GCD(m, k\%m)}{(k > m) \wedge (GCD(x, y) = GCD(m, k\%m))}$$

# Proof by Case Split

Perform a *case split.* Second case $(k = m)$:

$$\frac{(k = m) \wedge (GCD(x, y) = GCD(m, k)) \qquad GCD(m, m) = m}{(k = m) \wedge (GCD(x, y) = m)}$$

$$\frac{(k = m) \wedge (GCD(x, y) = m) \quad GCD(0, m) = m}{(k = m) \wedge (GCD(x, y) = GCD(0, m))}$$

$$\cfrac{\cfrac{(k = m) \wedge (GCD(x, y) = GCD(0, m)) \quad m\%m = 0}{(k = m) \wedge (GCD(x, y) = GCD(m\%m, m))}}{(k = m) \wedge (GCD(x, y) = GCD(k\%m, m))}$$

$$\frac{(k = m) \wedge (GCD(x, y) = GCD(k\%m, m)) \quad GCD(k\%m, m) = GCD(m, k\%m)}{(k = m) \wedge (GCD(x, y) = GCD(m, k\%m))}$$

# Proof by Case Split

Perform a *case split.* Second case $(k = m)$:

$$\frac{(k = m) \wedge (GCD(x, y) = GCD(m, k)) \qquad GCD(m, m) = m}{(k = m) \wedge (GCD(x, y) = m)}$$

$$\frac{(k = m) \wedge (GCD(x, y) = m) \quad GCD(0, m) = m}{(k = m) \wedge (GCD(x, y) = GCD(0, m))}$$

$$\frac{\dfrac{(k = m) \wedge (GCD(x, y) = GCD(0, m)) \quad m\%m = 0}{(k = m) \wedge (GCD(x, y) = GCD(m\%m, m))}}{(k = m) \wedge (GCD(x, y) = GCD(k\%m, m))}$$

$$\frac{(k = m) \wedge (GCD(x, y) = GCD(k\%m, m)) \quad GCD(k\%m, m) = GCD(m, k\%m)}{(k = m) \wedge (GCD(x, y) = GCD(m, k\%m))}$$

By combining the two cases we can conclude

$$GCD(x, y) = GCD(m, k\%m)$$

Note: Though tedious, this proof was still not entirely formal.
We *implicitly* applied a number of rules.

- Quantifier instantiation
- Transitivity of equality, substitution
- Following propositional rules:

$$\frac{P \wedge Q}{P} \qquad \frac{P \wedge Q}{Q} \qquad \frac{P \quad Q}{P \wedge Q} \qquad \frac{P \rightarrow Q}{\neg P \vee Q} \qquad \frac{\neg P \vee Q \quad P}{Q}$$

# Euclid's Algorithm – Loop Invariant

We have established:

$\texttt{while}\ (m \neq 0)\ \texttt{do}$
  $\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$
  r:=k%m; k:=m; m=r
  $\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$

Apply Hoare's loop rule:

$$\frac{\{P \wedge B\}\ C\ \{P\}}{\{P\}\ \text{while}\ B\ \text{do}\ C\ \{\neg B \wedge P\}}$$

with

- $P \stackrel{\text{def}}{=} (k \geq m) \wedge GCD(x, y) = GCD(k, m)$
- $B \stackrel{\text{def}}{=} (m \neq 0)$

# Euclid's Algorithm – Loop Invariant

We obtain:

$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$

```
while  (m ≠ 0) do
  r := k % m;
  k := m;
  m := r;
```

$\{\ (m = 0) \wedge (k \geq m) \wedge (GCD(x, y) = GCD(k, m))\}$

# Euclid's Algorithm – Loop Invariant

We obtain:

$$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$$

while $(m \neq 0)$ do

```
  r := k % m;
  k := m;
  m := r;
```

$$\{\ (m = 0) \wedge (k \geq m) \wedge (GCD(x, y) = GCD(k, m))\}$$

$$\dfrac{\dfrac{(m = 0) \wedge (GCD(x, y) = GCD(k, m))}{(GCD(x, y) = GCD(k, 0))} \qquad GCD(k, 0) = k}{GCD(x, y) = k}$$

# Euclid's Algorithm – Loop Invariant

We obtain:

$$\{(k \geq m) \wedge GCD(x,y) = GCD(k,m)\}$$

```
while  (m ≠ 0) do
  r := k % m;
  k := m;
  m := r;
```

$$\{\ GCD(x,y) = k\}$$

$$\frac{\dfrac{(m=0) \wedge (GCD(x,y) = GCD(k,m))}{(GCD(x,y) = GCD(k,0))} \quad GCD(k,0) = k}{GCD(x,y) = k}$$

# Euclid's Algorithm – Loop Invariant

- We have established

$$P \quad \stackrel{\text{def}}{=} \quad (k \geq m) \wedge GCD(x, y) = GCD(k, m)$$

  as a *loop invariant*
- If $P$ holds after $n$ iterations of the loop, it also holds after $n + 1$
- We still need to establish the *base case* $n = 0$

# Euclid's Algorithm – Induction, Base Case

Does

$$(k \geq m) \wedge GCD(x, y) = GCD(k, m)$$

hold at the beginning of the loop?

```
{?}
if (x > y) then
  k := x;
  m := y;
else
  k := y;
  m := x;
{(k ≥ m) ∧ GCD(x, y) = GCD(k, m)}
```

# Euclid's Algorithm – Induction, Base Case

$\{?\}$

```
if (x > y) then
  k := x;
  m := y;
else
  k := y;
  m := x;
```

$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$

## Euclid's Algorithm – Induction, Base Case

$\{?\}$

```
if (x > y) then
  k := x;
  m := y;
else
  k := y;
  m := x;
```

$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$

$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$

# Euclid's Algorithm – Induction, Base Case

$\{?\}$
`if` $(x > y)$ `then`

k := x;

m := y;

`else`

k := y;
$\{(k \geq x) \wedge GCD(x, y) = GCD(k, x)\}$
m := x;
$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$
$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$

# Euclid's Algorithm – Induction, Base Case

$\{?\}$

`if` $(x > y)$ `then`

  k := x;

  m := y;

`else`

  $\{(y \geq x) \wedge GCD(x, y) = GCD(y, x)\}$

  k := y;

  $\{(k \geq x) \wedge GCD(x, y) = GCD(k, x)\}$

  m := x;

  $\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$

$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$

# Euclid's Algorithm – Induction, Base Case

$\{?\}$
if $(x > y)$ then

k := x;

m := y;
$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$
else
$\{(y \geq x) \wedge GCD(x, y) = GCD(y, x)\}$
k := y;
$\{(k \geq x) \wedge GCD(x, y) = GCD(k, x)\}$
m := x;
$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$
$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$

# Euclid's Algorithm – Induction, Base Case

$\{?\}$
if $(x > y)$ then

k := x;
$\{(k \geq y) \wedge GCD(x, y) = GCD(k, y)\}$
m := y;
$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$
else
$\{(y \geq x) \wedge GCD(x, y) = GCD(y, x)\}$
k := y;
$\{(k \geq x) \wedge GCD(x, y) = GCD(k, x)\}$
m := x;
$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$
$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$

# Euclid's Algorithm – Induction, Base Case

$\{?\}$
`if` $(x > y)$ `then`
  $\{(x \geq y) \wedge GCD(x, y) = GCD(x, y)\}$
  k := x;
  $\{(k \geq y) \wedge GCD(x, y) = GCD(k, y)\}$
  m := y;
  $\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$
`else`
  $\{(y \geq x) \wedge GCD(x, y) = GCD(y, x)\}$
  k := y;
  $\{(k \geq x) \wedge GCD(x, y) = GCD(k, x)\}$
  m := x;
  $\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$
$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$

# Euclid's Algorithm – Induction, Base Case

Strengthen the pre-conditions of the branches:

- `then`-branch:

$$\dfrac{\dfrac{(x > y)}{(x \geq y)} \qquad (GCD(x,y) = GCD(x,y))}{(x \geq y) \wedge (GCD(x,y) = GCD(x,y))}$$

- `else`-branch:

$$\dfrac{\dfrac{\neg(x > y)}{(y \geq x)} \qquad \dfrac{(GCD(x,y) = GCD(x,y))}{(GCD(x,y) = GCD(y,x))}}{(y \geq x) \wedge (GCD(x,y) = GCD(y,x))}$$

# Euclid's Algorithm – Hoare's Conditional Rule

Apply

$$\frac{\{B \wedge P\}\ C_1\ \{Q\} \quad \{\neg B \wedge P\}\ C_2\ \{Q\}}{\{P\}\ \text{if}\ B\ \text{then}\ C_1\ \text{else}\ C_2\ \{Q\}}$$

with

- $B \stackrel{\text{def}}{=} (x > y)$
- $P \stackrel{\text{def}}{=} \text{true}$
- $Q \stackrel{\text{def}}{=} (k \geq m) \wedge GCD(x, y) = GCD(k, m)$
- $C_1 \stackrel{\text{def}}{=} k := x; m := y$
- $C_2 \stackrel{\text{def}}{=} k := y; m := x$

# Euclid's Algorithm – Hoare's Conditional Rule

We obtain

$\{\text{true}\}$
```
if (x > y) then
  k := x;
  m := y;
else
  k := y;
  m := x;
```
$\{(k \geq m) \wedge GCD(x, y) = GCD(k, m)\}$

# Euclid's Algorithm – Hoare's Compositional Rule

Finally:

$$\frac{\{P\}\ C_1\ \{Q\}\ ,\ \{Q\}\ C_2\ \{R\}}{\{P\}\ C_1\ ;\ C_2\ \{R\}}$$

where

$C_1 \stackrel{\text{def}}{=}$

```
if (x > y) then
  k := x;
  m := y;
else
  k := y;
  m := x;
```

$C_2 \stackrel{\text{def}}{=}$

```
while (m ≠ 0) do
  r := k % m;
  k := m;
  m := r;
```

and $P \stackrel{\text{def}}{=}$ true, $Q \stackrel{\text{def}}{=} (k \geq m) \wedge GCD(x, y) = GCD(k, m)$,
$R \stackrel{\text{def}}{=} (GCD(x, y) = k)$

## Euclid's Algorithm – Correctness Established

```
{true}
if (x > y) then
  k := x;
  m := y;
else
  k := y;
  m := x;
while (m ≠ 0) do
  r := k % m;
  k := m;
  m := r;
{GCD(x, y) = k}
```

# Disjunctive Invariants

How do we find an invariant for the following code fragment?

$\{$true$\}$

```
x = y = 0;
while (x ≠ 100) {
  if  (x ≥ 50) {
    y := y + 1;
  } else {
    skip;
  }
  x = x + 1;
}
```

$\{$y=50$\}$

# Disjunctive Invariants

How do we find an invariant for the following code fragment?

{true}
```
x = y = 0;
while (x ≠ 100) {
  if  (x ≥ 50) {
    y := y + 1;
  } else {
    skip;
  }
  x = x + 1;
}
```
{y=50}

Loop has two *phases*:

- ($x < 50$): x increases and y remains 0
- ($x \geq 50$): x and y increase in lockstep

# Disjunctive Invariants

```
if (x ≥ 50) {
  y := y + 1;
} else {
  skip;
}
x = x + 1;
```

Loop has two *phases*:

- $(x < 50) \wedge (y = 0)$
- $(x \geq 50) \wedge (x - y = 50)$

# Disjunctive Invariants

```
if  (x ≥ 50) {
  y := y + 1;
} else {
  skip;
}
x = x + 1;
```

Loop has two *phases*:

$$((x < 50) \wedge (\mathrm{y} = 0)) \quad \vee \quad ((x \geq 50) \wedge (\mathrm{x} - \mathrm{y} = 50))$$

# Disjunctive Invariants

```
if  (x ≥ 50) {

  y := y + 1;
} else {

  skip;
}

x = x + 1;
```

$\{((x < 50) \wedge (y = 0)) \ \vee \ ((x \geq 50) \wedge (x - y = 50))\}$

# Disjunctive Invariants

```
if  (x ≥ 50) {

  y := y + 1;
} else {

  skip;
}
```

$\{((x+1 < 50) \wedge (y = 0)) \; \vee \; ((x+1 \geq 50) \wedge (x+1-y = 50))\}$

```
x = x + 1;
```

$\{((x < 50) \wedge (y = 0)) \; \vee \; ((x \geq 50) \wedge (x - y = 50))\}$

# Disjunctive Invariants

if $(x \geq 50)$ {

```
  y := y + 1;
} else {

  skip;
}
```

$\{((x < 49) \wedge (y = 0)) \vee ((x \geq 49) \wedge (x + 1 - y = 50))\}$

```
x = x + 1;
```

$\{((x < 50) \wedge (y = 0)) \vee ((x \geq 50) \wedge (x - y = 50))\}$

# Disjunctive Invariants

if $(x \geq 50)$ {

  y := y + 1;
} else {
  $\{(x < 50) \wedge ((x < 49) \wedge (y = 0)) \vee ((x \geq 49) \wedge (x + 1 - y = 50))\}$
  skip;
}
$\{((x < 49) \wedge (y = 0)) \vee ((x \geq 49) \wedge (x + 1 - y = 50))\}$
x = x + 1;
$\{((x < 50) \wedge (y = 0)) \vee ((x \geq 50) \wedge (x - y = 50))\}$

# Disjunctive Invariants

```
if  (x ≥ 50) {


  y := y + 1;
} else {
  {((x < 49) ∧ (y = 0)) ∨ ((x = 49) ∧ (x − y = 49))}
  skip;
}
 {((x < 49) ∧ (y = 0)) ∨ ((x ≥ 49) ∧ (x + 1 − y = 50))}
x = x + 1;
{((x < 50) ∧ (y = 0)) ∨ ((x ≥ 50) ∧ (x − y = 50))}
```

# Disjunctive Invariants

$\texttt{if}\ \ (x \geq 50)\ \{$

$\{(\mathrm{x} \geq 50) \wedge$

$((\mathrm{x} < 49) \wedge (\mathrm{y} + 1 = 0)) \ \vee \ ((\mathrm{x} \geq 49) \wedge ((\mathrm{x} + 1) - (\mathrm{y} + 1) = 50))\}$

```
  y := y + 1;
} else {
```

$\{((\mathrm{x} < 49) \wedge (\mathrm{y} = 0)) \ \vee \ ((\mathrm{x} = 49) \wedge (\mathrm{x} - \mathrm{y} = 49))\}$

```
  skip;
}
```

$\{((x < 49) \wedge (\mathrm{y} = 0)) \ \vee \ ((\mathrm{x} \geq 49) \wedge (\mathrm{x} + 1 - \mathrm{y} = 50))\}$

```
x = x + 1;
```

$\{((\mathrm{x} < 50) \wedge (\mathrm{y} = 0)) \ \vee \ ((\mathrm{x} \geq 50) \wedge (\mathrm{x} - \mathrm{y} = 50))\}$

# Disjunctive Invariants

```
if (x ≥ 50) {

  {(x ≥ 50) ∧ (x − y = 50)}
  y := y + 1;
} else {
  {((x < 49) ∧ (y = 0)) ∨ ((x = 49) ∧ (x − y = 49))}
  skip;
}
{((x < 49) ∧ (y = 0)) ∨ ((x ≥ 49) ∧ (x + 1 − y = 50))}
x = x + 1;
{((x < 50) ∧ (y = 0)) ∨ ((x ≥ 50) ∧ (x − y = 50))}
```

# Disjunctive Invariants

```
if (x ≥ 50) {
  {(x ≥ 50) ∧ (x − y = 50)}
  y := y + 1;
} else {
  {((x < 49) ∧ (y = 0)) ∨ ((x = 49) ∧ (x − y = 49))}
  skip;
}
{((x < 49) ∧ (y = 0)) ∨ ((x ≥ 49) ∧ (x + 1 − y = 50))}
x = x + 1;
{((x < 50) ∧ (y = 0)) ∨ ((x ≥ 50) ∧ (x − y = 50))}
```

- We need to strengthen the preconditions of the branches!

# Disjunctive Invariants – Strengthen Branches

Then branch:

$$(\mathtt{x} \geq 50) \wedge ((\mathtt{x} < 50) \wedge (\mathtt{y} = 0)) \;\vee\; ((x \geq 50) \wedge (\mathtt{x} - \mathtt{y} = 50))$$

implies

$$(x \geq 50) \wedge (\mathtt{x} - \mathtt{y} = 50)$$

# Disjunctive Invariants – Strengthen Branches

Then branch:

$$(\mathtt{x} \geq 50) \wedge ((\mathtt{x} < 50) \wedge (\mathtt{y} = 0)) \;\vee\; ((x \geq 50) \wedge (\mathtt{x} - \mathtt{y} = 50))$$

implies

$$(x \geq 50) \wedge (\mathtt{x} - \mathtt{y} = 50)$$

Note that $(\mathtt{x} \geq 50) \wedge (\mathtt{x} < 50)$ is contradictory.

## Disjunctive Invariants – Strengthen Branches

Else branch:

$$(\mathtt{x} < 50) \wedge ((\mathtt{x} < 50) \wedge (\mathtt{y} = 0)) \;\vee\; ((x \geq 50) \wedge (\mathtt{x} - \mathtt{y} = 50))$$

implies

$$((\mathtt{x} < 49) \wedge (\mathtt{y} = 0)) \;\vee\; ((\mathtt{x} = 49) \wedge (\mathtt{x} - \mathtt{y} = 49))$$

# Disjunctive Invariants – Strengthen Branches

Else branch:

$$(\mathtt{x} < 50) \land ((\mathtt{x} < 50) \land (\mathtt{y} = 0)) \ \lor \ ((x \geq 50) \land (\mathtt{x} - \mathtt{y} = 50))$$

implies

$$((\mathtt{x} < 49) \land (\mathtt{y} = 0)) \ \lor \ ((\mathtt{x} = 49) \land (\mathtt{x} - \mathtt{y} = 49))$$

- Note that $(\mathtt{x} < 50) \land (\mathtt{x} \geq 50)$ is contradictory.

# Disjunctive Invariants – Strengthen Branches

Else branch:

$$(\mathtt{x} < 50) \wedge ((\mathtt{x} < 50) \wedge (\mathtt{y} = 0)) \;\vee\; ((x \geq 50) \wedge (\mathtt{x} - \mathtt{y} = 50))$$

implies

$$((\mathtt{x} < 49) \wedge (\mathtt{y} = 0)) \;\vee\; ((\mathtt{x} = 49) \wedge (\mathtt{x} - \mathtt{y} = 49))$$

- Note that $(\mathtt{x} < 50) \wedge (\mathtt{x} \geq 50)$ is contradictory.
- So we know that $\mathtt{y} = 0$ and $\mathtt{x} < 50$ . Case split over $\mathtt{x}$:
  - $\mathtt{x} < 49$. Then $(\mathtt{x} < 49) \wedge (\mathtt{y} = 0)$ holds.
  - $\mathtt{x} = 49$. Then $(\mathtt{x} = 49) \wedge (\mathtt{x} - \mathtt{y} = 49)$ holds.

# Disjunctive Invariants

We have established that the induction step for the invariant

$$((\mathtt{x} < 50) \wedge (\mathtt{y} = 0)) \;\vee\; ((x \geq 50) \wedge (\mathtt{x} - \mathtt{y} = 50))$$

works.
Does the base case work out?

## Disjunctive Invariants

We have established that the induction step for the invariant

$$((\mathtt{x} < 50) \wedge (\mathtt{y} = 0)) \ \vee \ ((x \geq 50) \wedge (\mathtt{x} - \mathtt{y} = 50))$$

works.

Does the base case work out?

- Yes, since $\mathtt{x} = \mathtt{y} = 0$.

# Disjunctive Invariants

That wasn't so hard, right?

# Disjunctive Invariants

That wasn't so hard, right?

- Remember: if-statements in a loop may require a *case split* (i.e., disjunction or implication) in the invariant.

## Hoare's Axioms: Summary

$$\frac{}{\{P[E/\mathtt{x}]\}\ \mathtt{x:=}E\ \{P\}}$$

$$\frac{\{P\}\ C_1\ \{Q\}\ ,\ \{Q\}\ C_2\ \{R\}}{\{P\}\ C_1\ ;\ C_2\ \{R\}}$$

$$\frac{\{B \wedge P\}\ C_1\ \{Q\}\quad \{\neg B \wedge P\}\ C_2\ \{Q\}}{\{P\}\ \text{if } B \text{ then } C_1 \text{ else } C_2\ \{Q\}}$$

$$\frac{P' \rightarrow P \quad \{P\}\ C\ \{Q\} \quad Q \rightarrow Q'}{\{P'\}\ C\ \{Q'\}}$$

$$\frac{\{P \wedge B\}\ C\ \{P\}}{\{P\}\ \text{while } B \text{ do } C\ \{\neg B \wedge P\}}$$