イロン イ理シ イヨン イヨン ヨー わらぐ

Program Verification using Hoare Logic - An Introduction -

Peter Heinig

Technical University of Munich

March 2007, JASS 2007

The Hoare Rules

Applications

function recursive(x,y)

メロト メポト メミト メミト しき しのへで

The Hoare Rules

イロン (得) イヨン イヨン ヨー のくで

function recursive(x,y) if x == 0 disp (y);

_

The Hoare Rules

Applications

イロン 不得 とくほう 不良 うくや

function recursive(x,y)
 if x == 0
 disp (y);
 else
 recursive(x-1,y+1);
 end

```
\begin{bmatrix} x \in \mathbb{N}_0 \land y \in \mathbb{Z} \end{bmatrix}
\underbrace{\begin{array}{c} \text{function recursive}(x, y) \\ \text{if } x == 0 \\ \text{disp } (y); \\ \text{else} \\ \text{recursive}(x-1, y+1); \\ \text{end} \\ \\ \begin{bmatrix} \text{If the program terminates, the value of } x + y \text{ gets printed.} \end{bmatrix}
```

$$\begin{bmatrix} x \in \mathbb{N}_0 \land y \in \mathbb{Z} \end{bmatrix}$$

$$\underbrace{function}_{if} recursive(x,y)$$

$$\underbrace{if}_{if} x == 0$$

$$\underbrace{disp}_{recursive(x-1,y+1);}$$

$$\underbrace{end}_{if}$$
[If the program terminates, the value of $x + y$ gets printed.]

How can we prove this assertion?

$$\begin{bmatrix} x \in \mathbb{N}_0 \land y \in \mathbb{Z} \end{bmatrix}$$

$$\underbrace{\begin{array}{c} \underline{function} & \text{recursive}(x, y) \\ \underline{if} & x == 0 \\ & \underline{disp} & (y); \\ \underline{else} \\ & \text{recursive}(x-1, y+1); \\ \underline{end} \\ \\ \begin{bmatrix} \text{If the program terminates, the value of } x + y \text{ gets printed.} \end{bmatrix}$$

How can we prove this assertion? Easy.

$$\begin{bmatrix} x \in \mathbb{N}_0 \land y \in \mathbb{Z} \end{bmatrix}$$

$$\underbrace{function}_{if} recursive(x,y)$$

$$\underbrace{if}_{aisp}_{else}(y);$$

$$\underbrace{else}_{recursive(x-1,y+1);}$$

$$\underbrace{end}_{end}$$
[If the program terminates, the value of $x + y$ gets printed.]

 $\begin{bmatrix} x \in \mathbb{N}_0 \land y \in \mathbb{Z} \end{bmatrix}$ $\underbrace{\text{function recursive}(x,y)}_{\substack{\text{if } x == 0 \\ \text{disp } (y); \\ else}}_{\substack{\text{recursive}(x-1,y+1); \\ end}}$ $\begin{bmatrix} \text{If the program terminates, the value of } x + y \text{ gets printed.} \end{bmatrix}$

Proof of Correctness: (Induction on the first argument)

 $x \in \mathbb{N}_0 \land y \in \mathbb{Z}$ function recursive(x,y) if x == 0disp (y); else recursive(x-1,y+1); end If the program terminates, the value of x + y gets printed. *Proof of Correctness:* (Induction on the first argument) If [x = 0], then $\forall y \in \mathbb{Z} : [\text{recursive}(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

 $x \in \mathbb{N}_0 \land y \in \mathbb{Z}$ function recursive(x,y) if x == 0disp (y); else recursive(x-1,y+1); end If the program terminates, the value of x + y gets printed. *Proof of Correctness:* (Induction on the first argument) If [x = 0], then $\forall y \in \mathbb{Z} : [recursive(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 < x \in \mathbb{N}_0$:

 $x \in \mathbb{N}_0 \land y \in \mathbb{Z}$ function recursive(x,y) if x == 0disp (y); else recursive(x-1,y+1); end If the program terminates, the value of x + y gets printed. *Proof of Correctness:* (Induction on the first argument) If |x = 0|, then $\forall y \in \mathbb{Z} : [\text{recursive}(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 < x \in \mathbb{N}_0$: $\forall y \in \mathbb{Z}$: [recursive(x,y)] \Rightarrow [x + y gets printed].

 $x \in \mathbb{N}_0 \land y \in \mathbb{Z}$ function recursive(x,y) if x == 0disp (y); else recursive(x-1,y+1); end If the program terminates, the value of x + y gets printed. *Proof of Correctness:* (Induction on the first argument) If |x = 0|, then $\forall y \in \mathbb{Z} : [\text{recursive}(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 < x \in \mathbb{N}_0$: $\forall y \in \mathbb{Z}$: [recursive(x,y)] \Rightarrow [x + y gets printed]. Then [recursive(x+1,y)]

 $x \in \mathbb{N}_0 \land y \in \mathbb{Z}$ function recursive(x,y) if x == 0disp (y); else recursive(x-1,y+1); end If the program terminates, the value of x + y gets printed. *Proof of Correctness:* (Induction on the first argument) If |x = 0|, then $\forall y \in \mathbb{Z} : [\text{recursive}(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 < x \in \mathbb{N}_0$: $\forall y \in \mathbb{Z} : [recursive(x,y)] \Rightarrow [x + y \text{ gets printed}].$ Then $[recursive(x+1,y)] \xrightarrow{Function Definition} [recursive(x,y+1)]$

 $x \in \mathbb{N}_0 \land y \in \mathbb{Z}$ function recursive(x,y) if x == 0disp (y); else recursive(x-1,y+1); end If the program terminates, the value of x + y gets printed. *Proof of Correctness:* (Induction on the first argument) If |x = 0|, then $\forall y \in \mathbb{Z} : [\text{recursive}(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 < x \in \mathbb{N}_0$: $\forall y \in \mathbb{Z} : [recursive(x,y)] \Rightarrow [x + y \text{ gets printed}].$ Then $[recursive(x+1,y)] \xrightarrow{Function Definition} [recursive(x,y+1)]$ Inductive Assumption [x + (y + 1) gets printed],

 $x \in \mathbb{N}_0 \land y \in \mathbb{Z}$ function recursive(x,y) if x == 0disp (y); else recursive(x-1,y+1); end If the program terminates, the value of x + y gets printed. *Proof of Correctness:* (Induction on the first argument) If |x = 0|, then $\forall y \in \mathbb{Z} : [\text{recursive}(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 < x \in \mathbb{N}_0$: $\forall y \in \mathbb{Z}$: [recursive(x,y)] \Rightarrow [x + y gets printed]. Then $[recursive(x+1,y)] \xrightarrow{Function Definition} [recursive(x,y+1)]$ Inductive Assumption [x + (y + 1) gets printed],and x + (y + 1) = (x + 1) + y.

 $x \in \mathbb{N}_0 \land y \in \mathbb{Z}$ function recursive(x,y) if x == 0disp (y); else recursive(x-1,y+1); end If the program terminates, the value of x + y gets printed. *Proof of Correctness:* (Induction on the first argument) If |x = 0|, then $\forall y \in \mathbb{Z} : [\text{recursive}(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 < x \in \mathbb{N}_0$: $\forall y \in \mathbb{Z}$: [recursive(x,y)] \Rightarrow [x + y gets printed]. Then $[recursive(x+1,y)] \xrightarrow{Function Definition} [recursive(x,y+1)]$ Inductive Assumption [x + (y + 1) gets printed],and x + (y + 1) = (x + 1) + y.

The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

function iterative(x,y) while x > 0x = x - 1;y = y+1;end disp (y);

イロン 不良 メイヨン イヨン ヨー のくや

```
function iterative(x,y)
while x > 0
    x = x-1;
    y = y+1;
end
disp (y);
```

```
\begin{bmatrix} x \in \mathbb{N}_0 \land y \in \mathbb{Z} \end{bmatrix}
<u>function</u> iterative(x,y)
<u>while</u> x > 0
x = x-1;
y = y+1;
<u>end</u>
<u>disp</u> (y);

[If the program terminates, the value of x + y gets printed.]
```

```
\begin{bmatrix} x \in \mathbb{N}_0 \land y \in \mathbb{Z} \end{bmatrix}
<u>function</u> iterative(x,y)
<u>while</u> x > 0
x = x-1;
y = y+1;
<u>end</u>
<u>disp</u> (y);

[If the program terminates, the value of x + y gets printed.]
```

How can we prove this assertion?

```
\begin{bmatrix} x \in \mathbb{N}_0 \land y \in \mathbb{Z} \end{bmatrix}
\underbrace{\text{function iterative}(x,y)}_{\substack{\text{while } x > 0 \\ x = x-1; \\ y = y+1; \\ \underbrace{\text{end}}_{\substack{\text{disp}}}(y); \\ \begin{bmatrix} \text{If the program terminates, the value of } x + y \text{ gets printed.} \end{bmatrix}
```

How can we prove this assertion? Easy?

```
\begin{bmatrix} x \in \mathbb{N}_0 \land y \in \mathbb{Z} \end{bmatrix}
\underbrace{function \text{ iterative}(x,y)}_{\substack{\text{while } x > 0 \\ x = x-1; \\ y = y+1; \\ \underbrace{end}_{\substack{\text{disp }}}(y); \\ \begin{bmatrix} \text{If the program terminates, the value of } x + y \text{ gets printed.} \end{bmatrix}
```

How can we prove this assertion? Easy? Lets try it again.

```
function iterative(x,y)
while x > 0
x = x-1;
y = y+1;
end
disp (y);
```



```
function iterative(x,y)
while x > 0
x = x-1;
y = y+1;
end
disp (y);
Proof of Correctness:
```



```
function iterative(x,y)
while x > 0
x = x-1;
y = y+1;
end
disp (y);
Proof of Correctness: (Induction on the first argument
```

```
function iterative(x,y)
while x > 0
x = x-1;
y = y+1;
end
disp (y);
Proof of Correctness: (Induction on the first argument ???)
```

 $\begin{array}{l} \underline{function} \mbox{ iterative}(x,y) \\ \underline{while} \ x > 0 \\ x = x-1; \\ y = y+1; \\ \underline{end} \\ \underline{disp} \ (y); \\ Proof \ of \ Correctness: \ (Induction \ on \ the \ first \ argument \ ???) \\ If \ [x = 0], \ then \\ \forall y \in \mathbb{Z}: \ [\ iterative(x,y)] \xrightarrow{Function \ Definition} \ [y = x + y \ gets \ printed]. \end{array}$

 $\begin{array}{l} \displaystyle \underbrace{ \text{function} \text{ iterative}(x,y) } \\ \displaystyle \underbrace{ \text{while } x > 0 } \\ \displaystyle x = x-1; \\ \displaystyle y = y+1; \\ \displaystyle \underbrace{ \text{end} } \\ \displaystyle \text{disp } (y); \end{array} \\ \displaystyle \underbrace{ \text{Proof of Correctness: (Induction on the first argument ???) } \\ \displaystyle \text{If } [x = 0], \text{ then} \\ \displaystyle \forall y \in \mathbb{Z}: \left[\text{iterative}(x,y) \right] \stackrel{\text{Function Definition}}{\Rightarrow} \left[y = x + y \text{ gets printed} \right]. \\ \displaystyle \text{Now assume that for some } 0 \leq x \in \mathbb{N}_0 : \end{array}$

function iterative(x,y) while x > 0x = x - 1;y = y+1;end disp (y); *Proof of Correctness:* (Induction on the first argument ???) If [x = 0], then $\forall y \in \mathbb{Z} : [iterative(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 \le x \in \mathbb{N}_0$: $\forall y \in \mathbb{Z}$: [iterative(x,y)] \Rightarrow [x + y gets printed].

function iterative(x,y) while x > 0x = x - 1;y = y+1;end disp (y); *Proof of Correctness:* (Induction on the first argument ???) If [x = 0], then $\forall y \in \mathbb{Z} : [iterative(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 \le x \in \mathbb{N}_0$: $\forall y \in \mathbb{Z}$: [iterative(x,y)] \Rightarrow [x + y gets printed]. Then |iterative(x+1,y)|

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

function iterative(x,y) while x > 0x = x - 1;y = y+1;end disp (y); *Proof of Correctness:* (Induction on the first argument ???) If [x = 0], then $\forall y \in \mathbb{Z} : [iterative(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 \le x \in \mathbb{N}_0$: $\forall y \in \mathbb{Z}$: [iterative(x,y)] \Rightarrow [x + y gets printed]. Then [iterative(x+1,y)] $\xrightarrow{\text{Function Definition}}$

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

function iterative(x,y) while x > 0x = x - 1;y = y+1;end disp (y); Proof of Correctness: (Induction on the first argument ???) If [x = 0], then $\forall y \in \mathbb{Z} : [iterative(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 \le x \in \mathbb{N}_0$: $\forall y \in \mathbb{Z}$: [iterative(x,y)] \Rightarrow [x + y gets printed]. Then [iterative(x+1,y)] $\xrightarrow{\text{Function Definition}}$

function iterative(x,y) while x > 0x = x - 1;v = v+1;end disp (y); *Proof of Correctness:* (Induction on the first argument ???) If [x = 0], then $\forall y \in \mathbb{Z} : [iterative(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 < x \in \mathbb{N}_0$: $\forall y \in \mathbb{Z}$: [iterative(x,y)] \Rightarrow [x + y gets printed]. Then $[iterative(x+1,y)] \xrightarrow{Function Definition} [iterative(x+1,y)]$

function iterative(x,y) while x > 0x = x - 1;y = y+1;end disp (y); *Proof of Correctness:* (Induction on the first argument ???) If [x = 0], then $\forall y \in \mathbb{Z} : [iterative(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 < x \in \mathbb{N}_0$: $\forall y \in \mathbb{Z}$: [iterative(x,y)] \Rightarrow [x + y gets printed]. Then $[iterative(x+1,y)] \xrightarrow{\text{Function Definition}} [iterative(x+1,y)]$ Useless.
function iterative(x,y) while x > 0x = x - 1;y = y+1;end disp (y); *Proof of Correctness:* (Induction on the first argument ???) If [x = 0], then $\forall y \in \mathbb{Z} : [iterative(x,y)] \xrightarrow{\text{Function Definition}} [y = x + y \text{ gets printed}].$ Now assume that for some $0 < x \in \mathbb{N}_0$: $\forall y \in \mathbb{Z} : [\text{iterative}(x,y)] \Rightarrow [x + y \text{ gets printed}].$ Then $[iterative(x+1,y)] \xrightarrow{\text{Function Definition}} [iterative(x+1,y)]$ Useless. We need new tools

The Hoare Rules

Applications

<ロ> <畳> <置> <置> < => < => <<</p>

The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

What is the problem?

• No expression replacement rule anymore.

The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

- No expression replacement rule anymore.
- Assignments.

The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

- No expression replacement rule anymore.
- Assignments.
- While loop.

The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

- No expression replacement rule anymore.
- Assignments.
- While loop.
- Variables exist in different states during execution.

The Hoare Rules

Applications

Adapting to the new situation:

```
\begin{bmatrix} x \in \mathbb{N}_0 \land y \in \mathbb{Z} \end{bmatrix}
<u>function</u> iterative(x,y)
<u>while</u> x > 0
x = x-1;
y = y+1;
<u>end</u>
<u>disp</u> (y);

[If the program terminates, the value of x + y gets printed.]
```

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

Taking care of states:

```
\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}
\underbrace{\text{function iterative}(x,y)}_{\left[x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z}\right]}
\underbrace{\text{while } x > 0}_{x = x-1;}
y = y+1;
\underbrace{\text{end}}_{\text{disp}}(y);
\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}
```

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

```
\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}
\underbrace{\text{function iterative}(x,y)}_{\left[x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z}\right]}
\underbrace{\text{while } x > 0}_{x = x-1;}
y = y+1;
\underbrace{\text{end}}_{\text{disp}}(y);
\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}
```

Clearly:

$$\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$$

$$\frac{\text{function iterative}(x,y)}{\begin{bmatrix} x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z} \end{bmatrix}} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{while } x > 0}{x = x - 1};$$

$$y = y + 1;$$

$$\frac{\text{end}}{\text{disp}} (y);$$

$$\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}$$

$$\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$$

$$\frac{\text{function} \text{ iterative}(x,y)}{\begin{bmatrix} x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z} \end{bmatrix}} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{while } x > 0}{\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix} \land x > 0}$$

$$x = x - 1;$$

$$y = y + 1;$$

$$\frac{\text{end}}{\text{disp}} (y);$$

$$\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}$$

The Hoare Rules

Applications

We claim that this is a loop invariant

$$\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$$

$$\frac{\text{function iterative}(x,y)}{\begin{bmatrix} x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z} \end{bmatrix}} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{while } x > 0}{\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix} \land x > 0}$$

$$x = x - 1;$$

$$y = y + 1;$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{end}}{\text{disp}} (y);$$

$$\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}$$

・ロシ ・母シ ・ヨシ ・ヨン ・ヨー うべつ

$$\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$$

$$\frac{\text{function iterative}(x,y)}{\begin{bmatrix} x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z} \end{bmatrix}} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{while } x > 0}{\begin{bmatrix} [x + y = A + B \land x \ge 0] \land x > 0 \end{bmatrix}}$$

$$x = x - 1;$$

$$\begin{bmatrix} x + (y + 1) = A + B \land x \ge 0 \end{bmatrix}$$

$$y = y + 1;$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{end}}{\text{disp}} (y);$$

$$\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}$$

$$\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$$

$$\frac{\text{function}}{\text{iterative}(x,y)} \begin{bmatrix} x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z} \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{while } x > 0}{\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix} \land x > 0 \end{bmatrix}} \begin{bmatrix} (x - 1) + (y + 1) = A + B \land (x - 1) \ge 0 \end{bmatrix}$$

$$x = x - 1;$$

$$\begin{bmatrix} x + (y + 1) = A + B \land x \ge 0 \end{bmatrix}$$

$$y = y + 1;$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{end}}{\text{disp}} (y);$$

$$\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}$$

$$\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$$

$$\frac{\text{function}}{\text{iterative}(x,y)} \begin{bmatrix} x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z} \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{while } x > 0}{\begin{bmatrix} (x + y = A + B \land x \ge 0] \land x > 0 \end{bmatrix}} \begin{bmatrix} (x - 1) + (y + 1) = A + B \land (x - 1) \ge 0 \end{bmatrix}$$

$$x = x - 1;$$

$$\begin{bmatrix} x + (y + 1) = A + B \land x \ge 0 \end{bmatrix}$$

$$y = y + 1;$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{end}}{\text{disp}} (y);$$

$$\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}$$

$$\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$$

$$\frac{\text{function}}{\text{function}} \text{ iterative}(x,y)$$

$$\begin{bmatrix} x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z} \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{while } x > 0}{\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix} \land x > 0 \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} (x - 1) + (y + 1) = A + B \land (x - 1) \ge 0 \end{bmatrix}$$

$$x = x - 1;$$

$$\begin{bmatrix} x + (y + 1) = A + B \land x \ge 0 \end{bmatrix}$$

$$y = y + 1;$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{end}}{\text{disp}} (y);$$

$$\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}$$

$$\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$$

$$\frac{\text{function} \text{ iterative}(x,y)}{\begin{bmatrix} x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z} \end{bmatrix}} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{while } x > 0}{\begin{bmatrix} [x + y = A + B \land x \ge 0] \land x > 0 \end{bmatrix}} \Rightarrow$$

$$\begin{bmatrix} (x - 1) + (y + 1) = A + B \land (x - 1) \ge 0 \end{bmatrix}$$

$$x = x - 1;$$

$$\begin{bmatrix} x + (y + 1) = A + B \land x \ge 0 \end{bmatrix}$$

$$y = y + 1;$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{end}}{\text{disp}} (y);$$

$$\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}$$

$$\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$$

$$\underbrace{\text{function iterative}(x,y)} \begin{bmatrix} x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z} \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\underbrace{\text{while } x > 0} \begin{bmatrix} [x + y = A + B \land x \ge 0] \land x > 0 \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} (x - 1) + (y + 1) = A + B \land (x - 1) \ge 0 \end{bmatrix}$$

$$x = x - 1;$$

$$\begin{bmatrix} x + (y + 1) = A + B \land x \ge 0 \end{bmatrix}$$

$$y = y + 1;$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\underbrace{\text{end}}{\text{disp}}(y);$$

$$\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}$$

$$\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$$

$$\frac{\text{function}}{\text{iterative}(x,y)} \begin{bmatrix} x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z} \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{while } x > 0}{\begin{bmatrix} [x + y = A + B \land x \ge 0] \land x > 0 \end{bmatrix}} \Rightarrow$$

$$\begin{bmatrix} (x - 1) + (y + 1) = A + B \land (x - 1) \ge 0 \end{bmatrix}$$

$$x = x - 1;$$

$$\begin{bmatrix} x + (y + 1) = A + B \land x \ge 0 \end{bmatrix}$$

$$y = y + 1;$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{end}}{\begin{bmatrix} [x + y = A + B \land x \ge 0] \end{cases}} \land \neg [x > 0] \end{bmatrix}$$

$$\frac{\text{disp}}{\begin{bmatrix} y \end{bmatrix}};$$
[If the program terminates, the value of $A + B$ gets printed.]

<ロ> <畳> <置> <置> < => < => <<</p>

$$\begin{bmatrix} A \in \mathbb{N}_{0} \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$$

$$\frac{\text{function}}{\text{function}} \text{ iterative}(x,y)$$

$$\begin{bmatrix} x = A \in \mathbb{N}_{0} \land y = B \in \mathbb{Z} \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{while } x > 0}{\begin{bmatrix} [x + y = A + B \land x \ge 0] \land x > 0 \end{bmatrix}} \Rightarrow$$

$$\begin{bmatrix} (x - 1) + (y + 1) = A + B \land (x - 1) \ge 0 \end{bmatrix}$$

$$x = x - 1;$$

$$\begin{bmatrix} x + (y + 1) = A + B \land x \ge 0 \end{bmatrix}$$

$$y = y + 1;$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\begin{bmatrix} end \\ [x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\begin{bmatrix} end \\ [x + y = A + B \land x \ge 0] \land \neg [x > 0] \end{bmatrix} \Rightarrow$$

$$\frac{\text{disp}}{\text{disp}} (y);$$
[If the program terminates, the value of $A + B$ gets printed.]

$$\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$$

$$\frac{\text{function}}{\text{iterative}(x,y)} \begin{bmatrix} x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z} \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{while } x > 0}{\begin{bmatrix} [x + y = A + B \land x \ge 0] \land x > 0 \end{bmatrix} \Rightarrow} \begin{bmatrix} (x - 1) + (y + 1) = A + B \land (x - 1) \ge 0 \end{bmatrix}$$

$$x = x - 1;$$

$$\begin{bmatrix} x + (y + 1) = A + B \land x \ge 0 \end{bmatrix}$$

$$y = y + 1;$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{end}}{\begin{bmatrix} [x + y = A + B \land x \ge 0] \land \neg [x > 0] \end{bmatrix} \Rightarrow} \begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\begin{bmatrix} \text{end} \\ [x + y = A + B \land x \ge 0] \land \neg [x > 0] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} \text{disp} (y); \end{bmatrix}$$
If the program terminates, the value of $A + B$ gets printed.

・ロン ・雪ン ・ヨン ・ヨー のへで

$$\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative } (A,B) \text{ is called.} \end{bmatrix}$$

$$\frac{\text{function}}{\text{iterative}(x,y)} \begin{bmatrix} x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z} \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{while } x > 0}{\begin{bmatrix} [x + y = A + B \land x \ge 0] \land x > 0 \end{bmatrix} \Rightarrow} \begin{bmatrix} (x - 1) + (y + 1) = A + B \land (x - 1) \ge 0 \end{bmatrix}$$

$$x = x - 1;$$

$$\begin{bmatrix} x + (y + 1) = A + B \land x \ge 0 \end{bmatrix}$$

$$y = y + 1;$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{end}}{\begin{bmatrix} [x + y = A + B \land x \ge 0] \land \neg [x > 0] \end{bmatrix} \Rightarrow} \begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix} \land \neg [x > 0] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix} \land \neg [x > 0] \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix} \Rightarrow \begin{bmatrix} y = A + B \end{bmatrix}$$

$$\frac{\text{disp}}{y} (y);$$

$$\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}$$

<ロ> <畳> <置> <置> < => < => <<</p>

$$\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$$

$$\frac{\text{function iterative}(x,y)}{\begin{bmatrix} x = A \in \mathbb{N}_0 \land y = B \in \mathbb{Z} \end{bmatrix} \Rightarrow}$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{while } x > 0}{\begin{bmatrix} [x + y = A + B \land x \ge 0] \land x > 0 \end{bmatrix} \Rightarrow}$$

$$\begin{bmatrix} (x - 1) + (y + 1) = A + B \land (x - 1) \ge 0 \end{bmatrix}$$

$$x = x - 1;$$

$$\begin{bmatrix} x + (y + 1) = A + B \land x \ge 0 \end{bmatrix}$$

$$y = y + 1;$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix}$$

$$\frac{\text{end}}{\begin{bmatrix} [x + y = A + B \land x \ge 0] \land \neg [x > 0] \end{bmatrix} \Rightarrow}$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix} \land \neg [x > 0] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix} \land \neg [x > 0] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} x + y = A + B \land x \ge 0 \end{bmatrix} \Rightarrow \begin{bmatrix} y = A + B \end{bmatrix}$$

$$\frac{\text{disp}}{y} (y);$$

$$\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}$$

The Hoare Rules

Applications

What have we done? What will we do?

The Hoare Rules

Applications

What have we done? What will we do?

The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

What have we done? What will we do?

• What, exactly, did we proof, after all? And what not?

The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

What have we done? What will we do?

- What, exactly, did we proof, after all? And what not?
- How can we codify what we have done and will have to do next time?

イロン 不得 とくほ とくほ とう うくや

What have we done? What will we do?

- What, exactly, did we proof, after all? And what not?
- How can we codify what we have done and will have to do next time?
- What are the underlying rules of reasoning?

What we did proof:

The Hoare Rules

Applications

What we did proof:

• Partial Semantic Correctness of the function iterative(x,y) with respect to some specification.



What we did proof:

- Partial Semantic Correctness of the function iterative(x,y) with respect to some specification.
- The specification was:

 $\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$ **PROGRAM** [If the program terminates, the value of A + B gets printed.]

What we did proof:

- Partial Semantic Correctness of the function iterative(x,y) with respect to some specification.
- The specification was: $\begin{bmatrix} A \in \mathbb{N}_0 \land B \in \mathbb{Z} \land \text{ iterative}(A,B) \text{ is called.} \end{bmatrix}$ PROGRAM $\begin{bmatrix} \text{If the program terminates, the value of } A + B \text{ gets printed.} \end{bmatrix}$
- Semantic denotes that we were concerned with the meaning of the program. We are not concerned with the syntax of the program.

The Hoare Rules

Applications



There does not happen anything contradicting the specification.

In particular, for partial correctness it is allowed that:

・ロト ・昼 ・ ・ 言 ・ ・ 言 ・ うへで

In particular, for partial correctness it is allowed that:

• The program never terminates.



イロン (得) イヨン イヨン ヨー のくで

In particular, for partial correctness it is allowed that:

- The program never terminates.
- The program does terminate, the specification is fulfilled, and something not to be found in the specification happens in addition to that.
In particular, for partial correctness it is allowed that:

- The program never terminates.
- The program does terminate, the specification is fulfilled, and something not to be found in the specification happens in addition to that.

Yet we will be content with partial correctness and will go on to concern ourselves with only that.

In particular, for partial correctness it is allowed that:

- The program never terminates.
- The program does terminate, the specification is fulfilled, and something not to be found in the specification happens in addition to that.

Yet we will be content with partial correctness and will go on to concern ourselves with only that.

• Total correctness means:

The program terminates, and there does not happen anything contradicting the specification. In particular, for partial correctness it is allowed that:

- The program never terminates.
- The program does terminate, the specification is fulfilled, and something not to be found in the specification happens in addition to that.

Yet we will be content with partial correctness and will go on to concern ourselves with only that.

• Total correctness means:

The program terminates, and there does not happen anything contradicting the specification.

• Proving that a program terminates can be hard.

The Hoare Rules

Applications

Codification of what we did: The Hoare Rules

• C. A. R. Hoare 1969:



Codification of what we did: The Hoare Rules

• C. A. R. Hoare 1969:

An Axiomatic Basis for Computer Programming

C. A. R. HOARE The Queen's University of Belfast,* Northern Ireland

In this paper an attempt is made to explore the <u>logical founda-</u> tions of computer programming by use of techniques which were first applied in the study of geometry and have later been extended to other branches of mathematics. This involves the elucidation of sets of axioms and <u>rules of inference</u> which can be used in proofs of the properties of computer programs.

Predicates

A predicate is a function from some set D to the set {true, false }:

$$P: D \rightarrow \{\texttt{true}, \texttt{false}\}$$

・ロン・4回シ ・ヨン・ヨン・ヨー・クタク

Strong and Weak

The Hoare Rules

Applications

The Hoare Rules

Applications

Strong and Weak

• By Definition $[A] \Rightarrow [B] :\Leftrightarrow \neg [A] \lor [B]$

イロン (理) (言) (言) 言) のくで

The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

Strong and Weak

By Definition

[A] ⇒ [B] :⇔ ¬[A] ∨ [B]

The predicate [false] is the strongest of all:

∀B : false ⇒ [B] ⇔ ¬false ∨ [B] ⇔ true ∨ [B] ⇔ true

The Hoare Rules

Applications

イロン 不得 とくほ とくほ とう うくや

Strong and Weak

By Definition

[A] ⇒ [B] :⇔ ¬[A] ∨ [B]

The predicate [false] is the strongest of all:

∀B: false ⇒ [B] ⇔ ¬false ∨ [B] ⇔ true ∨ [B] ⇔ true

The predicate [true] is the weakest of all:

∀B: true ⇒ [B] ⇔ ¬true ∨ [B] ⇔ false ∨ [B] ⇔
false ∨ [B] ⇔ [B]

The Hoare Rules

Applications

イロン 不得 とくほ とくほ とう うくや

Strong and Weak

By Definition $[A] \Rightarrow [B] \quad :\Leftrightarrow \quad \neg[A] \lor [B]$ • The predicate [false] is the strongest of all: $\forall B: \text{ false} \Rightarrow \begin{bmatrix} B \end{bmatrix} \Leftrightarrow \neg \text{false} \lor \begin{bmatrix} B \end{bmatrix}$ \Leftrightarrow true $\lor |B| \Leftrightarrow$ true • The predicate |true| is the weakest of all: $\forall B: \text{ true} \Rightarrow \begin{bmatrix} B \end{bmatrix} \Leftrightarrow \neg \text{true} \lor \begin{bmatrix} B \end{bmatrix} \Leftrightarrow$ false $\vee [B] \Leftrightarrow [B]$ • Thus:

 $\mathsf{false} \hspace{0.1 in} \Rightarrow \hspace{0.1 in} \cdots \hspace{0.1 in} \Rightarrow \hspace{0.1 in} \mathsf{true}$

The Hoare Rules

Applications

Strong and Weak

- By Definition $[A] \Rightarrow [B] \quad :\Leftrightarrow \quad \neg[A] \lor [B]$ • The predicate [false] is the strongest of all: $\forall B: \text{ false} \Rightarrow \begin{bmatrix} B \end{bmatrix} \Leftrightarrow \neg \text{false} \lor \begin{bmatrix} B \end{bmatrix}$ \Leftrightarrow true $\lor |B| \Leftrightarrow$ true • The predicate |true| is the weakest of all: $\forall B: \text{ true} \Rightarrow \begin{bmatrix} B \end{bmatrix} \Leftrightarrow \neg \text{true} \lor \begin{bmatrix} B \end{bmatrix} \Leftrightarrow$ false $\vee [B] \Leftrightarrow [B]$ • Thus:
- Thus:

 $\mathsf{false} \ \Rightarrow \ \cdots \ \Rightarrow \ \mathsf{true}$

• We define:

 $\begin{bmatrix} A \end{bmatrix} \text{ is stronger than } \begin{bmatrix} B \end{bmatrix} \quad :\Leftrightarrow \quad \begin{bmatrix} A \end{bmatrix} \Rightarrow \begin{bmatrix} B \end{bmatrix}$

The Hoare Rules

Applications

Strong and Weak

- By Definition $[A] \Rightarrow [B] \quad :\Leftrightarrow \quad \neg[A] \lor [B]$ • The predicate [false] is the strongest of all: $\forall B: \text{ false} \Rightarrow \begin{bmatrix} B \end{bmatrix} \Leftrightarrow \neg \text{false} \lor \begin{bmatrix} B \end{bmatrix}$ \Leftrightarrow true $\lor |B| \Leftrightarrow$ true • The predicate |true| is the weakest of all: $\forall B: \text{ true} \Rightarrow \begin{bmatrix} B \end{bmatrix} \Leftrightarrow \neg \text{true} \lor \begin{bmatrix} B \end{bmatrix} \Leftrightarrow$ false $\vee [B] \Leftrightarrow [B]$ • Thus:
- Thus:

 $\mathsf{false} \ \Rightarrow \ \cdots \ \Rightarrow \ \mathsf{true}$

• We define:

 $\begin{bmatrix} A \end{bmatrix} \text{ is stronger than } \begin{bmatrix} B \end{bmatrix} \quad :\Leftrightarrow \quad \begin{bmatrix} A \end{bmatrix} \Rightarrow \begin{bmatrix} B \end{bmatrix}$

The Hoare Rules

Applications

The Mathematical Structure of the Hoare Rules

- Essential ingredient: Hoare Triple:
- $\left[\left[\mathsf{P} \right] S \left[\mathsf{Q} \right] \right]$

イロン (得) イヨン イヨン ヨー のくで

The Hoare Rules

Applications

イロン 不得 とくほ とくほ とう うくや

- Essential ingredient: Hoare Triple: [P] S [Q]
- A Hoare Triple is itself a predicate $H: \{\texttt{true},\texttt{false}\} \times M \times \{\texttt{true},\texttt{false}\} \longrightarrow \{\texttt{true},\texttt{false}\},$

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

- Essential ingredient: Hoare Triple: [P] S [Q]
- A Hoare Triple is itself a predicate $H: \{\texttt{true},\texttt{false}\} \times M \times \{\texttt{true},\texttt{false}\} \longrightarrow \{\texttt{true},\texttt{false}\},$
- \bullet where the predicates $\left[P\right]$ and $\left[Q\right]$ provide the first and third argument, and

イロン 不得 とくほ とくほ とう うくや

- Essential ingredient: Hoare Triple: $\left[\left[P
 ight] S \left[Q
 ight]
 ight]$
- A Hoare Triple is itself a predicate $H: \{\texttt{true},\texttt{false}\} \times M \times \{\texttt{true},\texttt{false}\} \longrightarrow \{\texttt{true},\texttt{false}\},$
- \bullet where the predicates $\left[P\right]$ and $\left[Q\right]$ provide the first and third argument, and
- the set M denotes the set of all syntactically correct programs in some programming language,

- Essential ingredient: Hoare Triple: $\left[\left[P
 ight] S \left[Q
 ight]
 ight]$
- A Hoare Triple is itself a predicate $H: \{\texttt{true},\texttt{false}\} \times M \times \{\texttt{true},\texttt{false}\} \longrightarrow \{\texttt{true},\texttt{false}\},$
- where the predicates [P] and [Q] provide the first and third argument, and
- the set M denotes the set of all syntactically correct programs in some programming language,
- and the value of $\left[\left[P
 ight] S \left[Q
 ight]
 ight]$ is defined as follows:

The Hoare Rules

Applications

When is a Hoare Triple true, when is it false?

$$\begin{bmatrix} [P] & S & [Q] \end{bmatrix} = \texttt{true}$$

If the predicate [P] is true immediately before execution of the program $S \in M$, then immediately after S has terminated, the predicate [Q] is true.

The Hoare Rules

Applications

The rules often take the following form:

メロト メロト メミト メミト 三目 うえぐ

The Hoare Rules

Applications

イロン 不得 とくほ とくほ とう うくや

The rules often take the following form:

A formula involving predicates and Hoare Triples. A Hoare Triple whose program fragment comprises the fragments used in the Hoare Triples above.

The Hoare Rules

Applications

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

The rules often take the following form:



The Hoare Rules

Applications

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

The rules often take the following form:





The Hoare Rules

Applications

<ロ> <畳> <置> <置> < => < => <<</p>

Rule 0:

The Hoare Rules

Applications



The Hoare Rules

Applications

Rule 1: Axiom of Assignment



The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

Rule 1: Axiom of Assignment



The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

Rule 1: Axiom of Assignment



The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

Rule 2: Rule of Consequence

$$\frac{\left[\left[\left[\widetilde{P}\right]\Rightarrow\left[P\right]\right] \land \left[\left[P\right]S\left[q\right]\right] \land \left[\left[Q\right]\Rightarrow\left[\widetilde{q}\right]\right]\right]}{\left[\left[\widetilde{P}\right]S\left[\widetilde{q}\right]\right]}$$

The Hoare Rules

Applications

イロン 不得 とくほ とくほ とう うくや

Rule 3: Rule of Composition



The Hoare Rules

Applications

Rule 4: Rule of Iteration

$$\frac{\left[\left[P \land B\right] \quad \mathbf{S} \quad \left[P\right]\right]}{\left[\left[P\right] \quad \underline{\text{while } B \text{ do } \mathbf{S} \quad \underline{\text{end}} \quad \left[P \land \neg B\right]\right]}$$

The Hoare Rules

Applications

Rule 5: *Rule of Conditional Branching*

$$\frac{\left[\left[\left[P \land B \right] \quad \mathbf{S} \quad \left[Q \right] \right] \quad \wedge \quad \left[\left[P \land \neg B \right] \quad \mathbf{T} \quad \left[Q \right] \right] \right]}{\left[\left[P \right] \quad \underline{\text{if } B \text{ do } \mathbf{S} \quad \underline{\text{else}} \text{ do } \mathbf{T} \text{ end } \quad \left[Q \right] \right]}$$

Applications

The Hoare Rules

Applications

x = x + y; y = x - y;x = x - y;

The Hoare Rules

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$

x = x + y;
y = x - y;
x = x - y;

The Hoare Rules

Applications

Swapping without moving...

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$

$$\begin{array}{c} x = x + y; \\ y = x - y; \\ x = x - y; \\ \begin{bmatrix} x = B \land y = A \end{bmatrix} \end{array}$$

・ロン・4回シ ・ヨン・ヨン・ヨー・クタク
$$\left[x = A \land y = B\right]$$

$$x = x + y;$$

 $y = x - y;$
 $x = x - y;$

$$\left[x = B \land y = A\right]$$

Applications 0000●000000

Annotating the Program with Assertions

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$

$$\begin{bmatrix} x = x + y; \\ y = x - y; \\ x = x - y; \end{bmatrix}$$

$$\begin{bmatrix} x = B \land y = A \end{bmatrix}$$

・ロン ・聞 ・ ・ 言 と ・ 聞 と ・ の へ ()・

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$

$$\begin{bmatrix} x = x + y; \\ y = x - y; \\ x = x - y; \end{bmatrix}$$

$$\left\lfloor x=B \land y=A\right\rfloor$$

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$

$$\begin{bmatrix} x = x + y; \\ y = x - y; \\ x_{after} = x_{before} - y_{before} \end{bmatrix}$$

$$\begin{bmatrix} x_{after} = B \land y_{after} = A \end{bmatrix}$$

;

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$

$$\begin{bmatrix} x = x + y; \\ y = x - y; \\ x_{after} = x_{before} - y_{before}; \qquad y_{after} = y_{before};$$

 $\left[x_{\mathsf{after}} = B \land y_{\mathsf{after}} = A \right]$

F

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$

$$\begin{bmatrix} x = x + y; \\ y = x - y; \\ x_{before} - y_{before} = B \land y_{before} = A \\ x_{after} = x_{before} - y_{before}; \qquad y_{after} = y_{before}; \end{bmatrix}$$

-

$$\left[x_{\mathsf{after}} = B \land y_{\mathsf{after}} = A\right]$$

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$

$$\begin{bmatrix} x = x + y; \\ y = x - y; \\ x - y = B \land y = A \\ x = x - y; \end{bmatrix}$$

$$\left[x=B \land y=A\right]$$

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$

$$\begin{bmatrix} x = x + y; \\ y = x - y; \\ [x - y = B \land y = A] \\ x = x - y; \end{bmatrix}$$

$$\left\lfloor x=B \land y=A\right\rfloor$$

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$

$$x = x + y;$$

$$\begin{bmatrix} x - (x - y) = B \land x - y = A \end{bmatrix}$$

$$y = x - y;$$

$$\begin{bmatrix} x - y = B \land y = A \end{bmatrix}$$

$$x = x - y;$$

$$\left[x=B \land y=A\right]$$

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$

$$x = x + y;$$

$$\begin{bmatrix} x - (x - y) = B \land x - y = A \end{bmatrix}$$

$$y = x - y;$$

$$\begin{bmatrix} x - y = B \land y = A \end{bmatrix}$$

$$x = x - y;$$

$$\left[x=B \land y=A\right]$$

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$

$$x = x + y;$$

$$\begin{bmatrix} x - (x - y) = B \land x - y = A \end{bmatrix}$$

$$y = x - y;$$

$$\begin{bmatrix} x - y = B \land y = A \end{bmatrix}$$

$$x = x - y;$$

$$\left[x=B \land y=A\right]$$

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$x = x + y;$$

$$\begin{bmatrix} x - (x - y) = B \land x - y = A \\ y = x - y; \\ \begin{bmatrix} x - y = B \land y = A \\ x = x - y; \end{bmatrix}$$

$$\left[x=B \land y=A\right]$$

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$
$$\begin{bmatrix} (x+y) - ((x+y) - y) = B \land (x+y) - y = A \end{bmatrix}$$

$$x = x + y;$$

$$\begin{bmatrix} x - (x - y) = B \land x - y = A \end{bmatrix}$$

$$y = x - y;$$

$$\begin{bmatrix} x - y = B \land y = A \end{bmatrix}$$

$$x = x - y;$$

$$\left[x=B \land y=A\right]$$

$$\begin{bmatrix} x = A \land y = B \end{bmatrix}$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$x = x + y;$$

$$\begin{bmatrix} x - (x - y) = B \land x - y = A \end{bmatrix}$$

$$y = x - y;$$

$$\begin{bmatrix} x - y = B \land y = A \end{bmatrix}$$

$$x = x - y;$$

$$\left[x=B \land y=A\right]$$

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \Longrightarrow$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$x = x + y;$$

$$\begin{bmatrix} x - (x - y) = B \land x - y = A \end{bmatrix}$$

$$y = x - y;$$

$$\begin{bmatrix} x - y = B \land y = A \end{bmatrix}$$

$$x = x - y;$$

$$\left[x=B \land y=A\right]$$

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \Longrightarrow$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$x = x + y;$$

$$\begin{bmatrix} x - (x - y) = B \land x - y = A \end{bmatrix}$$

$$y = x - y;$$

$$\begin{bmatrix} x - y = B \land y = A \end{bmatrix}$$

$$x = x - y;$$

$$\left[x=B \land y=A\right]$$

Applications

Using the Hoare Rules

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \Longrightarrow$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$x = x + y;$$

$$\begin{bmatrix} x - (x - y) = B \land x - y = A \end{bmatrix}$$

$$y = x - y;$$

$$\begin{bmatrix} x - y = B \land y = A \end{bmatrix}$$

$$x = x - y;$$

$$\left[x=B \land y=A\right]$$

Applications

Using the Hoare Rules

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \Longrightarrow$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$x = x + y;$$

$$\begin{bmatrix} x - (x - y) = B \land x - y = A \end{bmatrix}$$

$$y = x - y;$$

$$\begin{bmatrix} x - y = B \land y = A \end{bmatrix}$$

$$x = x - y;$$

$$\left[x=B \land y=A\right]$$

The Hoare Rules

Applications

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \Longrightarrow$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$\begin{array}{l} \mathbf{x} = \mathbf{x} + \mathbf{y}; \\ \left[x - (x - y) = B \ \land \ x - y = A \right] \\ \mathbf{y} = \mathbf{x} - \mathbf{y}; \\ \left[x - y = B \ \land \ y = A \right] & \text{We appeal to Rule 2: Rule of Consequence} \\ \mathbf{x} = \mathbf{x} - \mathbf{y}; & \frac{\left[\left[\tilde{P} \right] \Rightarrow \left[P \right] \ \land \ \left[P \right] \begin{array}{c} \mathbf{S} \ \left[Q \right] \ \land \ \left[Q \right] \Rightarrow \left[\tilde{Q} \right] \right] \\ & \left[\left[\tilde{P} \right] \begin{array}{c} \mathbf{S} \ \left[\tilde{Q} \right] \end{array} \right] \end{array}$$

$$\left[x=B \land y=A\right]$$

The Hoare Rules

Applications

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \Longrightarrow$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$\begin{array}{l} x = x + y; \\ \left[x - (x - y) = B \ \land \ x - y = A \right] \\ y = x - y; \\ \left[x - y = B \ \land \ y = A \right] & \text{We analyze its constituent parts:} \\ x = x - y; & \displaystyle \frac{ \left[\left[\widetilde{P} \right] \Rightarrow \left[P \right] \ \land \ \left[P \right] \begin{array}{c} S \ \left[Q \right] \ \land \ \left[Q \right] \Rightarrow \left[\widetilde{Q} \right] \right] \\ & \left[\left[\widetilde{P} \right] \begin{array}{c} S \ \left[\widetilde{Q} \right] \end{array} \right] \end{array}$$

$$\left[x=B \land y=A\right]$$

The Hoare Rules

Applications

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \Longrightarrow$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$\begin{array}{l} x = x + y; \\ \left[x - (x - y) = B \ \land \ x - y = A \right] \\ y = x - y; \\ \left[x - y = B \ \land \ y = A \right] & \text{Here the rule is applicable:} \\ x = x - y; & \frac{ \left[\left[\widetilde{P} \right] \Rightarrow \left[P \right] \ \land \ \left[P \right] \begin{array}{c} S \ \left[Q \right] \ \land \ \left[Q \right] \Rightarrow \left[\widetilde{Q} \right] \right] \\ & \left[\left[\widetilde{P} \right] \begin{array}{c} S \ \left[\widetilde{Q} \right] \end{array} \right] \end{array}$$

$$\left[x=B \land y=A\right]$$

The Hoare Rules

Applications

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \Longrightarrow$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$\begin{array}{l} \mathbf{x} = \mathbf{x} + \mathbf{y}; \\ \begin{bmatrix} x - (x - y) = B \ \land \ x - y = A \end{bmatrix} \\ \mathbf{y} = \mathbf{x} - \mathbf{y}; \\ \begin{bmatrix} x - y = B \ \land \ y = A \end{bmatrix} & \text{We analyze its constituent parts:} \\ \mathbf{x} = \mathbf{x} - \mathbf{y}; & \frac{\left[\begin{bmatrix} \tilde{P} \end{bmatrix} \Rightarrow \begin{bmatrix} P \end{bmatrix} \ \land \ \begin{bmatrix} P \end{bmatrix} \begin{array}{c} \mathbf{S} \ \begin{bmatrix} Q \end{bmatrix} \ \land \ \begin{bmatrix} Q \end{bmatrix} \Rightarrow \begin{bmatrix} \tilde{Q} \end{bmatrix} \right] \\ \begin{bmatrix} \begin{bmatrix} \tilde{P} \end{bmatrix} \begin{array}{c} \mathbf{S} \ \begin{bmatrix} \tilde{Q} \end{bmatrix} \end{bmatrix} \end{array}$$

$$\left[x=B \land y=A\right]$$

Applications

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \Longrightarrow$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$\begin{array}{l} \mathbf{x} = \mathbf{x} + \mathbf{y}; \\ \left[x - (x - y) = B \ \land \ x - y = A \right] \\ \mathbf{y} = \mathbf{x} - \mathbf{y}; \\ \left[x - y = B \ \land \ y = A \right] & \text{We appeal to Rule 1: Axiom of Assignment} \\ \mathbf{x} = \mathbf{x} - \mathbf{y}; & \frac{\left[\left[\widetilde{P} \right] \Rightarrow \left[P \right] \ \land \ \left[P \right] \begin{array}{c} \mathbf{S} \ \left[Q \right] \ \land \ \left[Q \right] \Rightarrow \left[\widetilde{Q} \right] \right] \\ & \left[\left[\widetilde{P} \right] \begin{array}{c} \mathbf{S} \ \left[\widetilde{Q} \right] \end{array} \right] \end{array}$$

$$\left[x=B \land y=A\right]$$

Applications

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \Longrightarrow$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$\begin{array}{l} \mathbf{x} = \mathbf{x} + \mathbf{y}; \\ \begin{bmatrix} x - (x - y) = B \ \land \ x - y = A \end{bmatrix} \\ \mathbf{y} = \mathbf{x} - \mathbf{y}; \\ \begin{bmatrix} x - y = B \ \land \ y = A \end{bmatrix} & \text{Here we have to see an implication.} \\ \mathbf{x} = \mathbf{x} - \mathbf{y}; & \frac{\left[\begin{bmatrix} \tilde{P} \end{bmatrix} \Rightarrow \begin{bmatrix} P \end{bmatrix} \ \land \ \begin{bmatrix} P \end{bmatrix} \begin{array}{c} \mathbf{S} \begin{bmatrix} Q \end{bmatrix} \ \land \ \begin{bmatrix} Q \end{bmatrix} \Rightarrow \begin{bmatrix} \tilde{Q} \end{bmatrix} \right] \\ \begin{bmatrix} \begin{bmatrix} \tilde{P} \end{bmatrix} \begin{array}{c} \mathbf{S} \begin{bmatrix} \tilde{Q} \end{bmatrix} \end{bmatrix} \end{array}$$

$$\left[x=B \land y=A\right]$$

The Hoare Rules

Applications

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \Longrightarrow$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$\begin{array}{l} \mathbf{x} = \mathbf{x} + \mathbf{y}; \\ \begin{bmatrix} x - (x - y) = B \ \land \ x - y = A \end{bmatrix} \\ \mathbf{y} = \mathbf{x} - \mathbf{y}; \\ \begin{bmatrix} x - y = B \ \land \ y = A \end{bmatrix} & \text{And there actually is one; let } \widetilde{Q} = Q. \\ \mathbf{x} = \mathbf{x} - \mathbf{y}; & \frac{\left[\begin{bmatrix} \widetilde{P} \end{bmatrix} \Rightarrow \begin{bmatrix} P \end{bmatrix} \ \land \ \begin{bmatrix} P \end{bmatrix} \begin{array}{c} \mathbf{S} \ \begin{bmatrix} Q \end{bmatrix} \ \land \ \begin{bmatrix} Q \end{bmatrix} \Rightarrow \begin{bmatrix} Q \end{bmatrix} \right] \\ \hline \begin{bmatrix} \begin{bmatrix} \widetilde{P} \end{bmatrix} \begin{array}{c} \mathbf{S} \ \begin{bmatrix} Q \end{bmatrix} \end{bmatrix} \end{array}$$

$$\left[x=B \land y=A\right]$$

The Hoare Rules

Applications

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \implies$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$\begin{array}{l} \mathbf{x} = \mathbf{x} + \mathbf{y}; \\ \left[x - (x - y) = B \ \land \ x - y = A \right] \\ \mathbf{y} = \mathbf{x} - \mathbf{y}; \\ \left[x - y = B \ \land \ y = A \right] \quad \text{Thus } \dots \\ \mathbf{x} = \mathbf{x} - \mathbf{y}; \quad \frac{\left[\left[\widetilde{P} \right] \Rightarrow \left[P \right] \ \land \ \left[P \right] \ \mathbf{S} \left[\mathbf{Q} \right] \ \land \ \left[\mathbf{Q} \right] \Rightarrow \left[\widetilde{\mathbf{Q}} \right] \right] \\ \left[\left[\widetilde{P} \right] \ \mathbf{S} \left[\widetilde{\mathbf{Q}} \right] \right] \end{array}$$

$$\left[x=B \land y=A\right]$$

The Hoare Rules

Applications

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \implies$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$\begin{array}{l} x = x + y; \\ \left[x - (x - y) = B \ \land \ x - y = A \right] \\ y = x - y; \\ \left[x - y = B \ \land \ y = A \right] \quad \text{Thus } ... \\ x = x - y; \quad \displaystyle \frac{ \left[\left[\widetilde{P} \right] \Rightarrow \left[P \right] \ \land \ \left[P \right] \begin{array}{c} S \ \left[Q \right] \ \land \ \left[Q \right] \Rightarrow \left[\widetilde{Q} \right] \right] \\ \end{array} \right] \\ \left[\left[\widetilde{P} \right] \begin{array}{c} S \ \left[\widetilde{Q} \right] \end{array} \right] \end{array}$$

$$\left[x=B \land y=A\right]$$

The Hoare Rules

Applications

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \implies$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$\begin{array}{l} x = x + y; \\ \left[x - (x - y) = B \ \land \ x - y = A \right] \\ y = x - y; \\ \left[x - y = B \ \land \ y = A \right] & \text{And one step in the program is proved.} \\ x = x - y; & \displaystyle \frac{ \left[\left[\widetilde{P} \right] \Rightarrow \left[P \right] \ \land \ \left[P \right] \begin{array}{c} S \ \left[Q \right] \ \land \ \left[Q \right] \Rightarrow \left[\widetilde{Q} \right] \right] \\ \left[\left[\widetilde{P} \right] \begin{array}{c} S \ \left[\widetilde{Q} \right] \end{array} \right] \end{array}$$

$$\left[x=B \land y=A\right]$$

The Hoare Rules

Applications

$$\begin{bmatrix} x = A \land y = B \end{bmatrix} \implies$$
$$\begin{bmatrix} (x + y) - ((x + y) - y) = B \land (x + y) - y = A \end{bmatrix}$$

$$\begin{array}{l} \mathbf{x} = \mathbf{x} + \mathbf{y}; \\ \begin{bmatrix} x - (x - y) = B \ \land \ x - y = A \end{bmatrix} \\ \mathbf{y} = \mathbf{x} - \mathbf{y}; \\ \begin{bmatrix} x - y = B \ \land \ y = A \end{bmatrix} & \text{We could go on like that } \dots \\ \mathbf{x} = \mathbf{x} - \mathbf{y}; & \frac{\left[\left[\widetilde{P} \right] \Rightarrow \left[P \right] \ \land \ \left[P \right] \begin{array}{c} \mathbf{S} \ \left[Q \right] \\ \left[\widetilde{P} \right] \end{bmatrix} \right] \\ \hline \begin{bmatrix} \left[\widetilde{P} \right] \begin{array}{c} \mathbf{S} \ \left[\widetilde{Q} \right] \end{bmatrix} \end{array}$$

$$\left[x=B \land y=A\right]$$

Applications

Finding suitable invariants may be not that easy.



The Hoare Rules

Applications

Symmetry helps.

The Hoare Rules

Applications

<u>function</u> result = f(x,y)

Applications

Symmetry helps.

 $\frac{\text{function}}{\text{while}} \text{ x < y || y < x}$



Symmetry helps.

The Hoare Rules

Applications

イロン 不得 システン 不良 シスク

$\frac{\text{function}}{\text{while } x < y || y < x}$ $\frac{\text{if } x < y}{\text{if } x < y}$

Applications

Symmetry helps.



Applications

イロン (得) イヨン イヨン ヨー のくで

Symmetry helps.

function result = f(x,y)
while x < y || y < x
if x < y
 y = y-x;
else
 x = x-y;
end
end
result =</pre>
Applications

イロン (得) イヨン イヨン ヨー のくで

Symmetry helps.

function result = f(x,y)
while x < y || y < x
if y < x
x = x-y;
else
y = y-x;
end
end
result =</pre>

The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

What shall be our result?

```
function result = f(x,y)
while x < y || y < x
if x < y
    y = y-x;
else
    x = x-y;
end
end
result =</pre>
```

The Hoare Rules

Applications

What shall be our result?

```
function result = f(x,y)
while x < y || y < x
if x < y
y = y-x;
else
x = x-y;
end
end
result =
> Skip proof that x=y=gcd(A,B)
```



The Hoare Rules

Applications

Proof that x = y = gcd(A, B)

function result = f(x,y)
while x < y || y < x
if x < y
y = y-x;
else
x = x-y;
end
end</pre>

・ロト ・ 聞 ト ・ 言 ト ・ 言 ト ・ の へ (で)

イロン (得) イヨン イヨン ヨー のくで

```
\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}
\frac{\text{function}}{\text{while } x < y | | y < x}
\frac{\text{if } x < y}{y = y - x};
\frac{\text{else}}{x = x - y};
\frac{\text{end}}{\text{end}}
[If the program terminates, x = y = \text{gcd}(A, B)]
```

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(\mathbb{A}, \mathbb{B}) \text{ is called.}] \end{bmatrix}$$

$$\frac{\text{function result} = f(x, y)$$

$$\begin{bmatrix} x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\frac{\text{while } x < y \mid | y < x}{\underset{y = y-x;}{\underset{end}{end}}}$$

$$\begin{bmatrix} \text{else} \\ x = x-y; \\ \\ end \\ \end{bmatrix}$$

$$\begin{bmatrix} \text{If the program terminates, } x = y = \gcd(A, B) \end{bmatrix}$$

Applications

Proof that x = y = gcd(A, B)

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\underline{function result} = f(x,y)$$

$$\begin{bmatrix} [x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \\ \underline{while} \ x < y \ | | \ y < x \\ \underline{if} \ x < y \\ y = y-x; \\ \underline{else} \\ x = x-y; \\ \underline{end} \\ \underline{end} \\ \end{bmatrix}$$

$$\begin{bmatrix} else \\ nd \end{bmatrix}$$

$$\begin{bmatrix} end \\ end \\ \end{bmatrix}$$

$$\begin{bmatrix} ft he program terminates, \ x = y = gcd(A,B) \end{bmatrix}$$

The Hoare Rules

Applications

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\frac{\text{function result} = f(x,y)$$

$$\begin{bmatrix} [x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\frac{\text{while } x < y || y < x}{\left[[gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \right]}$$

$$\frac{\text{if } x < y}{y = y - x};$$

$$\frac{\text{else}}{x = x - y};$$

$$\frac{\text{end}}{\left[[gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \right]}$$

$$\begin{bmatrix} \text{end}}{\left[[gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \right]}$$

$$\begin{bmatrix} \text{end}}{\left[[gcd(x,y) = gcd(A,B)] \land \neg [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \right]}$$

$$\begin{bmatrix} \text{If the program terminates, } x = y = gcd(A,B) \end{bmatrix}$$

The Hoare Rules

Applications

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\underline{function} \text{ result} = f(x,y)$$

$$\begin{bmatrix} [x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{while} x < y || y < x$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{if} x < y$$

$$y = y - x;$$

$$\underline{else}$$

$$x = x - y;$$

$$\underline{end}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{end}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

The Hoare Rules

Applications

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\underline{function} \text{ result} = f(x,y)$$

$$\begin{bmatrix} [x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{while} x < y || y < x$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{if} x < y$$

$$y = y - x;$$

$$\underline{else}$$

$$x = x - y;$$

$$\underline{end}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{end}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

The Hoare Rules

Applications

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\underline{function} \text{ result} = f(x,y)$$

$$\begin{bmatrix} [x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{while} x < y || y < x$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{if} x < y$$

$$y = y - x;$$

$$\underline{else}$$

$$x = x - y;$$

$$\underline{end}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{end}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

The Hoare Rules

Applications

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\frac{\text{function result} = f(x,y)$$

$$\begin{bmatrix} [x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\frac{\text{while } x < y \ | | y < x$$

$$\begin{bmatrix} gcd(x,y) = gcd(A,B) \end{bmatrix} \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\frac{\text{if } x < y}{y = y - x;}$$

$$\frac{\text{else}}{x = x - y;}$$

$$\frac{\text{end}}{\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}}$$

$$\begin{bmatrix} \text{end} \\ [gcd(x,y) = gcd(A,B) \end{bmatrix} \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} \text{end} \\ [gcd(x,y) = gcd(A,B) \end{bmatrix} \land \neg [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} \text{If the program terminates, } x = y = gcd(A,B) \end{bmatrix}$$

Applications

Proof that x = y = gcd(A, B)

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\underline{function} \text{ result} = f(x,y)$$

$$\begin{bmatrix} [x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{while} x < y \mid | y < x$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{if} x < y$$

$$y = y - x;$$

$$\underline{else}$$

$$x = x - y;$$

$$\underline{else}$$

$$x = x - y;$$

$$\underline{end}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{end}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B)] \land \neg [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} If the program terminates, x = y = gcd(A,B) \end{bmatrix}$$

Applications

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\underline{function} \text{ result} = f(x,y)$$

$$\begin{bmatrix} [x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{while} x < y | | y < x$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$P$$

$$\underline{if} x < y$$

$$y = y - x;$$

$$\underline{else}$$

$$x = x - y;$$

$$\underline{end}$$

$$\begin{bmatrix} Q \\ \\ [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{end}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ \\ [gcd(x,y) = gcd(A,B)] \land \neg [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} If \text{ the program terminates, } x = y = gcd(A,B) \end{bmatrix}$$

Applications

Proof that x = y = gcd(A, B)

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\frac{\text{function}}{\text{function}} \text{ result} = f(x,y)$$

$$\begin{bmatrix} [x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\frac{\text{while } x < y | | y < x}{\left[[gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \right]}$$

$$\text{Let } \begin{bmatrix} P \end{bmatrix} := \begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\frac{\text{if } x < y}{y = y - x};$$

$$\frac{\text{else}}{x = x - y};$$

$$\frac{\text{end}}{x}$$

$$\text{Let } \begin{bmatrix} Q \end{bmatrix} := \begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} gcd(x,y) = gcd(A,B) \end{bmatrix} \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B) \end{bmatrix} \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} end \\ [gcd(x,y) = gcd(A,B) \end{bmatrix} \land \neg [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} If \text{ the program terminates, } x = y = gcd(A,B) \end{bmatrix}$$

Applications

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\frac{\text{function}}{\text{result}} \text{ result} = f(x,y) \\ [[x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1]] \Rightarrow \\ [[gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1]] \\ \frac{\text{while } x < y \mid y < x}{[[gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1]]} \\ \text{Let } [P] := [[gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1]] \\ \frac{\text{if } x < y}{y = y - x}; \\ \frac{else}{x = x - y}; \\ \frac{end}{d} \\ \text{Let } [Q] := [[gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1]] \\ [[gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1]] \\ [[gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1]] \\ [[gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1]] \\ [[fcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1]] \\ [If the program terminates, x = y = gcd(A,B)]$$

Applications

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\frac{\text{function}}{\text{function}} \text{ result} = f(x,y)$$

$$\begin{bmatrix} [x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\frac{\text{while}}{\text{is } x < y || y < x}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} P = \left[[gcd(x,y) = gcd(A,B) \right] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \right]$$

$$\frac{\text{if } x < y}{y = y - x};$$

$$\frac{\text{else}}{x = x - y};$$

$$\frac{\text{end}}{(Q) = \left[[gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \right]$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \right]$$

$$\begin{bmatrix} gcd(x,y) = gcd(A,B) \end{bmatrix} \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} gcd(x,y) = gcd(A,B) \end{bmatrix} \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B) \end{bmatrix} \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} If \text{ the program terminates, } x = y = gcd(A,B) \end{bmatrix}$$

The inner block

$$[P] = \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \ge 1 \right] \land \left[\mathbb{N} \ni y \ge 1 \right] \right]$$

$$\begin{array}{r} \underbrace{if \ x < y} \\ y = y - x; \\ \underbrace{else} \\ x = x - y; \\ \left[Q \right] = \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \ge 1 \right] \land \left[\mathbb{N} \ni y \ge 1 \right] \right] \end{array}$$

Applications

The Rule of Conditional Branching demands:

$$\begin{bmatrix} P \end{bmatrix} = \begin{bmatrix} [\gcd(x, y) = \gcd(A, B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$
$$\begin{bmatrix} \underline{if} \ x < y \\ [[\gcd(x, y) = \gcd(A, B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \land [x < y] \end{bmatrix}$$

$$y = y-x;$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\frac{else}{[gcd(x,y) = gcd(A,B)] \land [x \ne y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \land \neg [x < y] \end{bmatrix}$$

$$\begin{array}{l} \mathbf{x} = \mathbf{x} - \mathbf{y}; \\ \left[\left[\mathsf{gcd}(\mathbf{x}, \mathbf{y}) = \mathsf{gcd}(\mathbf{A}, \mathbf{B}) \right] \land \left[\mathbb{N} \ni \mathbf{x} \ge 1 \right] \land \left[\mathbb{N} \ni \mathbf{y} \ge 1 \right] \right] \\ \\ \underline{\mathsf{end}} \\ \left[\mathbf{Q} \right] = \left[\left[\mathsf{gcd}(\mathbf{x}, \mathbf{y}) = \mathsf{gcd}(\mathbf{A}, \mathbf{B}) \right] \land \left[\mathbb{N} \ni \mathbf{x} \ge 1 \right] \land \left[\mathbb{N} \ni \mathbf{y} \ge 1 \right] \right] \end{array}$$

Applications

By the Axiom of Assignment:

$$\begin{split} & \left[P\right] = \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[x \neq y\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ & \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[x \neq y\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right] \land \left[x < y\right]\right] \\ & \left[\left[\gcd(x,y-x) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y - x \geq 1\right]\right] \\ & y = y \cdot x; \\ & \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ & \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[x \neq y\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right] \land \neg[x < y]\right] \\ & \left[\left[\gcd(x-y,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x - y \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ & x = x \cdot y; \\ & \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ & \left[Q\right] = \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \end{aligned}$$

There are the implications:

$$\begin{split} & [P] = \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \frac{\mathrm{if}}{\mathrm{if}} x < y \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \land \left[x < y \right] \right] \Rightarrow \\ & \left[\left[\gcd(x, y - x) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y - x \geq 1 \right] \right] \\ & y = y - x; \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \frac{\mathrm{else}}{\left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \land \neg [x < y] \right] \Rightarrow \\ & \left[\left[\gcd(x - y, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x - y \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & x = x - y; \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[Q \right] = \left[\left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \end{aligned}$$

$$\begin{split} \left[P\right] &= \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[x \neq y\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ &= \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[x \neq y\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right] \land \left[x < y\right]\right] \Rightarrow \\ \left[\left[\gcd(x,y-x) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y - x \geq 1\right]\right] \\ &= y = y - x; \\ \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ &= \frac{else}{e} \\ \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[x \neq y\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right] \land \neg [x < y]\right] \Rightarrow \\ \left[\left[\gcd(x-y,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x - y \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ &= x = y; \\ \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ \left[Q\right] &= \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \end{split}$$

The Hoare Rules

Applications

Where:

<u>if B do</u> S <u>else</u> do T <u>end</u>

$$\left[Q\right]$$

$$\begin{split} & \left[P\right] = \left\lfloor \left[\gcd(x,y) = \gcd(A,B)\right] \land \left[x \neq y\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right] \right] \\ & \underbrace{\texttt{if}} x < y \\ & \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[x \neq y\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right] \land \left[x < y\right]\right] \Rightarrow \\ & \left[\left[\gcd(x,y-x) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y - x \geq 1\right]\right] \\ & y = y - x; \\ & \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ & \underbrace{\texttt{else}} \\ & \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[x \neq y\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right] \land \neg[x < y]\right] \Rightarrow \\ & \left[\left[\gcd(x-y,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x - y \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ & x = x - y; \\ & \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ & \left[\operatorname{end} \\ & \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \end{aligned}$$

The Hoare Rules

Applications

Where: [P] if B do S else do T end [Q]

$$\begin{split} & \left[P\right] = \begin{bmatrix} \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \land \left[x < y \right] \right] \Rightarrow \\ & \left[\left[\gcd(x, y - x) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y - x \geq 1 \right] \right] \\ & y = y - x; \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x - y \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x - y \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & x = x - y; \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[Q\right] = \left[\left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \end{aligned}$$

<u>do</u> T <u>end</u> [Q]

 $\backslash \Lambda$

$$\begin{split} & \left[P\right] = \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \land \left[x < y \right] \right] \Rightarrow \\ & \left[\left[\gcd(x, y - x) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y - x \geq 1 \right] \right] \\ & y = y - x; \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x - y \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & x = x - y; \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[Q\right] = \left[\left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \end{aligned}$$

Applications

The Hoare Rules

Applications

Where: $[P] \quad \underline{if} \; B \; \underline{do} \mathbf{S} \quad \underline{else} \; \underline{do} \; \mathbf{T} \; \underline{end} \; [Q]$

$$\begin{split} & [P] = \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \underbrace{\texttt{if} \ x < y} \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \land \left[x < y \right] \right] \Rightarrow \\ & \left[\left[\gcd(x, y - x) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y - x \geq 1 \right] \right] \\ & y = y - x; \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x - y, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x - y \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & x = x - y; \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[Q \right] = \left[\left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \end{aligned}$$

・ロン・4回シ ・ヨン・ヨン・ヨー・クタク

The Hoare Rules

Applications

Where: $[P] \quad \underline{if} B \underline{do} S \underline{else} \underline{do} \quad \boxed{P} \underline{end} \quad [Q]$

$$\begin{split} & \left[P\right] = \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \land \left[x < y \right] \right] \Rightarrow \\ & \left[\left[\gcd(x, y - x) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y - x \geq 1 \right] \right] \\ & y = y - x; \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x - y \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x - y \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \end{aligned}$$

The Hoare Rules

Where: $[P] \quad \underline{if} \; B \; \underline{do} S \quad \underline{else} \; \underline{do} \; T \; \underline{end}$

$$\begin{split} & [P] = \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \underbrace{\texttt{if}} x < y \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \land \left[x < y \right] \right] \Rightarrow \\ & \left[\left[\gcd(x, y - x) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y - x \geq 1 \right] \right] \\ & y = y - x; \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[x \neq y \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x - y, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x - y \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & x = x - y; \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \\ & \left[Q \right] = \left[\left[\left[\gcd(x, y) = \gcd(A, B) \right] \land \left[\mathbb{N} \ni x \geq 1 \right] \land \left[\mathbb{N} \ni y \geq 1 \right] \right] \end{aligned}$$

Applications

So the inner block is proved.

$$\begin{split} & \left[P\right] = \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[x \neq y\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ & \frac{\mathrm{if} \ x < y}{\left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[x \neq y\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right] \land \left[x < y\right]\right] \Rightarrow} \\ & \left[\left[\gcd(x,y-x) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y - x \geq 1\right]\right] \\ & y = y - x; \\ & \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ & \frac{\mathrm{else}}{\left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[x \neq y\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right] \land \neg[x < y]\right] \Rightarrow} \\ & \left[\left[\gcd(x-y,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x - y \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ & x = x - y; \\ & \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \\ & \left[Q\right] = \left[\left[\gcd(x,y) = \gcd(A,B)\right] \land \left[\mathbb{N} \ni x \geq 1\right] \land \left[\mathbb{N} \ni y \geq 1\right]\right] \end{aligned}$$

Applications

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\underline{function} \text{ result} = f(x,y)$$

$$\begin{bmatrix} [x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\underline{while} x < y | | y < x$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} P \end{bmatrix} = \begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} P \end{bmatrix} = \begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} P \end{bmatrix} = \begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} P \end{bmatrix} \begin{bmatrix} gcd(x,y) = gcd(A,B) \end{bmatrix} \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} gcd(x,y) = gcd(A,B) \end{bmatrix} \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} gcd(x,y) = gcd(A,B) \end{bmatrix} \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} gcd(x,y) = gcd(A,B) \end{bmatrix} \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} gcd(x,y) = gcd(A,B) \end{bmatrix} \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} fthe program terminates, x = y = gcd(A,B) \end{bmatrix}$$

・ロト ・ 聞 ト ・ 言 ト ・ 言 ・ うへぐ

Applications

Thus, in our main proof:

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\frac{\text{function}}{\text{function}} \text{ result} = f(x,y)$$

$$\begin{bmatrix} [x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\frac{\text{while}}{\text{is } x < y || y < x}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} P = \left[[gcd(x,y) = gcd(A,B) \right] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \right]$$

$$\frac{\text{if } x < y}{y = y - x;}$$

$$\frac{\text{else}}{x = x - y;}$$

$$\frac{\text{end}}{(Q] = \left[[gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \right]$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \right]$$

$$\begin{bmatrix} gcd(x,y) = gcd(A,B) \end{bmatrix} \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} gcd(x,y) = gcd(A,B) \end{bmatrix} \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B) \end{bmatrix} \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} fthe program terminates, x = y = gcd(A,B) \end{bmatrix}$$

Applications

There are three collisions left; trivially:

$$\begin{bmatrix} [\mathbb{N} \ni A \ge 1] \land [\mathbb{N} \ni B \ge 1] \land [f(A,B) \text{ is called.}] \end{bmatrix}$$

$$\frac{\text{function}}{\text{function}} \text{ result} = f(x,y)$$

$$\begin{bmatrix} [x = A] \land [y = B] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\frac{\text{while}}{\text{[gcd}(x,y) = gcd(A,B)} \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} P \end{bmatrix} = \begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\frac{\text{if } x < y}{y = y - x;}$$

$$\frac{\text{else}}{x = x - y;}$$

$$\frac{\text{end}}{(Q] = \begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} [gcd(x,y) = gcd(A,B)] \land \neg [x \neq y] \land [\mathbb{N} \ni x \ge 1] \land [\mathbb{N} \ni y \ge 1] \end{bmatrix}$$

$$\begin{bmatrix} If \text{ the program terminates, } x = y = gcd(A,B) \end{bmatrix}$$

Applications

And, clearly:

<ロ> <置> <置> < 置> < 置> < 置 > のQ()

The Hoare Rules

Applications

What shall be our result?

The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

What shall be our result?

```
function result = f(x,y)
while x < y || y < x
if x < y
y = y-x;
else
x = x-y;
end
end
result = (x+y)/2;</pre>
```

Applications

What we know:

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
 function result = f(x,y)
    while x < y \mid \mid y < x
       if x < y
          y = y - x;
       else
          x = x - y;
       end
    end
    result = (x+y)/2;
If the program terminates, \texttt{result} = \texttt{gcd}(A, B)
```

・ロン ・雪 と ・言 と ・ 言 ・ うへの
Applications

Let us make it more symmetric.

The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

Only one result is rather asymmetric...

Applications

Only one result is rather asymmetric...

Applications

More results may need more variables...

Applications

イロン (得) イヨン イヨン ヨー のくで

More results may need more variables...

```
<u>function</u> [fR, sR] = f(x,y)
  u = x;
  v = y;
  while x < y \mid \mid y < x
    if x < y
      y = y - x;
    else
      x = x - y;
    end
  end
  fR = (x+y)/2;
  sR = ?
```

Applications

イロン (得) イヨン イヨン ヨー のくで

We had better balance this surplus of minuses...

```
function [fR, sR] = f(x, y)
 u = x;
 v = y;
 while x < y \mid | y < x
    if x < y
      y = y - x;
      v = v+u;
    else
      x = x - y;
    end
 end
  fR = (x+y)/2;
  sR = ?
```

Applications

We had better balance this surplus of minuses...

```
function [fR, sR] = f(x, y)
  u = x;
  v = y;
  while x < y \mid \mid y < x
    if x < y
      y = y - x;
      v = v+u;
    else
      x = x - y;
      u = u + v;
    end
  end
  fR = (x+y)/2;
  sR = ?
```

Applications

What shall our second result be?

<u>function</u> [fR, sR] = $f(x, y)$
u = x;
v = y;
<u>while</u> x < y y < x
<u>if</u> x < y
y = y - x;
v = v+u;
else
x = x - y;
u = u + v;
end
<u>end</u>
fR = (x+y)/2;
sR = ?

The Hoare Rules

Applications

イロン (得) イヨン イヨン ヨー のくで

A symmetric one, of course...

```
function [fR, sR] = f(x, y)
 u = x;
 v = y;
 while x < y \mid | y < x
    if x < y
      y = y - x;
      v = v+u;
    else
      x = x - y;
      u = u + v;
    end
 end
  fR = (x+y)/2;
  sR = (u+v)/2;
```

Applications

So, what is sR?

```
function [fR, sR] = f(x, y)
  u = x;
  v = y;
  while x < y \mid \mid y < x
    if x < y
      y = y - x;
      v = v+u;
    else
      x = x - y;
      u = u + v;
    end
  end
  fR = (x+y)/2;
  sR = (u+v)/2;
```

Applications

Well, what is the counterpart to gcd?

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
 function [fR, sR] = f(x, y)
    u = x:
    v = y;
    while x < y \mid \mid y < x
      if x < y
         y = y - x;
         v = v+u:
       else
         x = x - y;
         u = u + v:
       end
    end
    fR = (x+y)/2;
    sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
```

Applications

How could we find a good invariant?

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
 function [fR, sR] = f(x, y)
    u = x:
    v = y;
    while x < y \mid \mid y < x
      if x < y
         y = y - x;
         v = v+u:
       else
         x = x - y;
         u = u + v:
       end
    end
    fR = (x+y)/2;
    sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
```

The Hoare Rules

Applications

We use what we know...

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
 function [fR, sR] = f(x, y)
    u = x;
    v = y;
    while x < y \mid \mid y < x
      if x < y
         y = y - x;
         v = v+u:
       else
         x = x - y;
         u = u + v:
       end
    end
    fR = (x+y)/2;
    sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

Applications

イロン イ理シ イヨン イヨン ヨー のくで

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
 function [fR, sR] = f(x,y)
                                 Well, suppose we are right in both assertions.
    u = x:
    v = y;
    while x < y \mid \mid y < x
      if x < y
         y = y - x;
         v = v+u:
      else
         x = x - y;
         u = u + v:
      end
    end
    fR = (x+y)/2;
    sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

Applications

イロン イ理シ イヨン イヨン ヨー のくで

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
 function [fR, sR] = f(x,y)
                                 Well, suppose we are right in both assertions.
    u = x:
                                Then, upon termination,
    v = y;
    while x < y \mid \mid y < x
      if x < y
         y = y - x;
         v = v+u:
      else
         x = x - y;
         u = u + v:
      end
    end
    fR = (x+y)/2;
    sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
 function [fR, sR] = f(x,y)
                                 Well, suppose we are right in both assertions.
    u = x:
                                 Then, upon termination,
    v = y;
    while x < y \mid \mid y < x
      if x < y
                                              gcd(A, B) \cdot scm(A, B) = fR \cdot sR
         y = y - x;
         v = v+u:
       else
         x = x - y;
         u = u + v:
       end
    end
    fR = (x+y)/2;
    sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
 function [fR, sR] = f(x,y)
                                    Well, suppose we are right in both assertions.
    u = x:
                                   Then, upon termination,
    v = y;
    while x < y \mid \mid y < x
       if x < y
                                            A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
          y = y - x;
          v = v+u:
       else
          x = x - y;
          u = u + v:
       end
    end
    fR = (x+y)/2;
    sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

...and try to invent a loop invariant.

```
\mathbb{N} \ni A \geq 1 \ \land \ \mathbb{N} \ni B \geq 1 \ \land \ \mathtt{f}(\mathtt{A},\mathtt{B}) is called.
  function [fR, sR] = f(x,y)
                                        Well, suppose we are right in both assertions.
     u = x:
                                        Then, upon termination,
     v = v;
     while x < y \mid \mid y < x
        if x < y
                                                 A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                                But: \frac{x+y}{2} \cdot \frac{u+y}{2} = fR \cdot sR
           v = v+u:
        else
           x = x - y;
           u = u + v;
        end
     end
     fR = (x+y)/2;
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

(ロ) (四) (三) (三) (三) (三) (三) (○) (○)

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x,y)
                                     Well, suppose we are right in both assertions.
    u = x:
                                     Then, upon termination,
     v = v;
    while x < y \mid \mid y < x
       if x < y
                                              A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
          y = y - x;
                                                Upon termination \frac{x+x}{2} \cdot \frac{u+v}{2} = fR \cdot sR
          v = v+u:
       else
          x = x - y;
          u = u + v:
       end
     end
     fR = (x+y)/2;
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x,y)
                                     Well, suppose we are right in both assertions.
    u = x:
                                     Then, upon termination,
     v = y;
    while x < y \mid \mid y < x
       if x < y
                                              A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
          y = y - x;
                                                Upon termination: \frac{x \cdot u + x \cdot v}{2} = fR \cdot sR
          v = v+u:
       else
          x = x - y;
          u = u + v:
       end
     end
     fR = (x+y)/2;
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x,y)
                                     Well, suppose we are right in both assertions.
    u = x:
                                     Then, upon termination,
     v = y;
    while x < y \mid \mid y < x
       if x < y
                                              A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
          y = y - x;
                                                Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
          v = v+u:
       else
          x = x - y;
          u = u + v:
       end
     end
     fR = (x+y)/2;
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

...and try to invent a loop invariant.

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x,y)
                                       Well, suppose we are right in both assertions.
     u = x:
                                       Then, upon termination,
     v = v;
     while x < y \mid \mid y < x
        if x < y
                                                A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                   Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
           v = v+u:
        else
                                                   Together: A \cdot B = \frac{y \cdot u + x \cdot v}{2}
           x = x - y;
                                                   Or: 2 \cdot A \cdot B = v \cdot u + x \cdot v
           u = u + v:
        end
     end
     fR = (x+y)/2;
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

・ロン ・ 理 > ・ モン ・ モー ・ のへぐ

Might $2 \cdot A \cdot B = y \cdot u + x \cdot v$ also be true during execution?

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x,y)
                                       We conjecture that this is a loop invariant.
     u = x;
     v = v;
     while x < y \mid \mid y < x
        if x < y
                                                A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                   Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
           v = v+u:
        else
                                                   Together: A \cdot B = \frac{y \cdot u + x \cdot v}{2}
           x = x - y;
                                                   Or: 2 \cdot A \cdot B = v \cdot u + x \cdot v
           u = u + v:
        end
     end
     fR = (x+y)/2;
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

・ロン ・ 理 > ・ モン ・ モー ・ のへぐ

The Hoare Rules

Applications

イロン イ理シ イヨン イヨン ヨー のくで

Yes.

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x,y)
                                       We conjecture that this is a loop invariant.
     u = x:
     v = v;
     while x < y \mid \mid y < x
        if x < y
                                                A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                   Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
           v = v+u:
        else
                                                   Together: A \cdot B = \frac{y \cdot u + x \cdot v}{2}
           x = x - y;
                                                   Or: 2 \cdot A \cdot B = v \cdot u + x \cdot v
           u = u + v:
        end
     end
     fR = (x+y)/2;
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

The Hoare Rules

Applications

Yes.

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x,y)
                                      We conjecture that this is a loop invariant.
    u = x:
     v = v;
    while x < y \mid \mid y < x
       if x < y
                                               A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                  Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
           v = v+u:
        else
                                                  Together: A \cdot B = \frac{y \cdot u + x \cdot v}{2}
           x = x - y;
                                                  Or: 2 \cdot A \cdot B = v \cdot u + x \cdot v
           u = u + v:
        end
     end
                                                 Upon initialization,
     fR = (x+y)/2;
                                                  x = u = A and y = v = B
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

The Hoare Rules

Applications

Yes.

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  <u>function</u> [fR, sR] = f(x,y)
                                          We conjecture that this is a loop invariant.
     u = x:
     v = v;
     while x < y \mid \mid y < x
        if x < y
                                                   A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                       Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
           v = v+u:
        else
                                                      Together: A \cdot B = \frac{y \cdot u + x \cdot v}{2}
           x = x - y;
                                                       Or 2 \cdot A \cdot B = v \cdot \mu + x \cdot v
           u = u + v:
        end
     end
                                                     So.
                                             2 \cdot A \cdot B = y \cdot u + x \cdot v \Leftrightarrow 2 \cdot A \cdot B = 2 \cdot A \cdot B
     fR = (x+y)/2;
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

イロン (四) イヨン (ヨン) ヨー のへの

The Hoare Rules

Applications

Yes.

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x,y)
                                       We conjecture that this is a loop invariant.
     u = x:
     v = v;
     while x < y \mid \mid y < x
        if x < y
                                                 A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                   Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
           v = v+u:
        else
                                                   Together: A \cdot B = \frac{y \cdot u + x \cdot v}{2}
           x = x - y;
                                                   Or: 2 \cdot A \cdot B = v \cdot u + x \cdot v
           u = u + v:
        end
     end
                                                  In the one branch,
                                          2 \cdot A \cdot B = y \cdot u + x \cdot v becomes
     fR = (x+y)/2;
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

Yes.

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x, y)
                                        We conjecture that this is a loop invariant.
     u = x;
     v = v;
     while x < y \mid \mid y < x
        if x < y
                                                 A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                    Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
           v = v+u:
        else
                                                    Together: A \cdot B = \frac{y \cdot u + x \cdot v}{2}
           x = x - y;
                                                    Or: 2 \cdot A \cdot B = y \cdot u + x \cdot v
           u = u + v:
        end
                                                   In the one branch
     end
                                           2 \cdot A \cdot B = y \cdot u + x \cdot y becomes
     fR = (x+y)/2;
     sR = (u+v)/2:
                                          2 \cdot A \cdot B = (y - x) \cdot u + x \cdot (v + u)
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

The Hoare Rules

Applications

Yes.

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x,y)
                                       We conjecture that this is a loop invariant.
     u = x:
     v = v;
     while x < y \mid \mid y < x
        if x < y
                                                A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                   Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
           v = v+u:
        else
                                                   Together: A \cdot B = \frac{y \cdot u + x \cdot v}{2}
           x = x - y;
                                                   Or: 2 \cdot A \cdot B = v \cdot u + x \cdot v
           u = u + v:
        end
     end
                                                  In the other branch,
                                          2 \cdot A \cdot B = y \cdot u + x \cdot v becomes
     fR = (x+y)/2;
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

・ロト ・日本 ・ヨト ・ヨト ・日 ・ のへで

Yes.

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x, y)
                                        We conjecture that this is a loop invariant.
     u = x;
     v = v;
     while x < y \mid \mid y < x
        if x < y
                                                 A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                    Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
           v = v+u:
        else
                                                    Together: A \cdot B = \frac{y \cdot u + x \cdot v}{2}
           x = x - y;
                                                    Or: 2 \cdot A \cdot B = y \cdot u + x \cdot v
           u = u + v:
        end
                                                   In the other branch.
     end
                                           2 \cdot A \cdot B = y \cdot u + x \cdot y becomes
     fR = (x+y)/2;
     sR = (u+v)/2:
                                          2 \cdot A \cdot B = y \cdot (u + v) + (x - y) \cdot v
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

・ロト ・ 通 ト ・ 言 ト ・ 言 ・ うへで

The Hoare Rules

Applications

Yes.

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x,y)
                                       We conjecture that this is a loop invariant.
     u = x:
     v = v;
     while x < y \mid \mid y < x
       if x < y
                                                A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                   Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
           v = v+u:
        else
                                                  Together: A \cdot B = \frac{y \cdot u + x \cdot v}{2}
           x = x - y;
                                                   Or: 2 \cdot A \cdot B = v \cdot u + x \cdot v
           u = u + v:
        end
     end
                                                 Thus, the equality is maintained.
     fR = (x+y)/2;
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

・ロト ・日本 ・ヨト ・ヨト ・ヨー うへで

The Hoare Rules

Applications

Yes.

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x,y)
                                      We conjecture that this is a loop invariant.
    u = x:
     v = v;
    while x < y \mid \mid y < x
       if x < y
                                               A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                  Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
           v = v+u:
        else
                                                  Together: A \cdot B = \frac{y \cdot u + x \cdot v}{2}
           x = x - y;
                                                  Or: 2 \cdot A \cdot B = v \cdot u + x \cdot v
           u = u + v:
        end
     end
                                                 Upon completion,
     fR = (x+y)/2;
                                        x = y = \gcd(A, B), thus
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

・ロト ・昼 ト ・言 ト ・ 言 ・ の々で

Yes.

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x, y)
                                      We conjecture that this is a loop invariant.
    u = x;
     v = v;
    while x < y \mid \mid y < x
       if x < y
                                               A \cdot B = \operatorname{gcd}(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                  Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
           v = v+u:
        else
                                                  Together: A \cdot B = \frac{y \cdot u + x \cdot v}{2}
           x = x - y;
                                                  Or: 2 \cdot A \cdot B = y \cdot u + x \cdot v
           u = u + v:
        end
                                                 Upon completion,
     end
                           x = y = \gcd(A, B), thus
     fR = (x+y)/2;
     sR = (u+v)/2:
                                         2 \cdot A \cdot B = y \cdot u + x \cdot v = \gcd(A, B) \cdot (u + v)
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B)
```

The Hoare Rules

Applications

Yes.

```
\mathbb{N} \ni A \ge 1 \land \mathbb{N} \ni B \ge 1 \land f(A,B) is called.
  function [fR, sR] = f(x, y)
                                        We conjecture that this is a loop invariant.
     u = x:
     v = v;
     while x < y \mid \mid y < x
        if x < y
                                                  A \cdot B = \gcd(A, B) \cdot \operatorname{scm}(A, B) = fR \cdot sR
           y = y - x;
                                                     Upon termination: \frac{y \cdot u + x \cdot v}{2} = fR \cdot sR
           v = v+u:
        else
                                                     Together: A \cdot B = \frac{y \cdot u + x \cdot v}{2}
Or: 2 \cdot A \cdot B = y \cdot u + x \cdot v
           x = x - y;
           u = u + v:
        end
     end
     fR = (x+y)/2;
                                           Therefore
                                           sR = \frac{u+v}{2} = \frac{A \cdot B}{gcd(A,B)} = scm(A,B)
     sR = (u+v)/2;
If the program terminates, sR = scm(A, B).
If the program terminates, fR = gcd(A, B).
```

イロン (四) イヨン (ヨン) ヨー のへの