Programming Languages: Imperative Program Construction
Practicals 4: Hoare Logic and Weakest Precondition: Loop

Shin-Cheng Mu

Autumn Term, 2022

1. Prove the correctness of the following program:

$$\begin{array}{l}\textbf{con}\ N : \mathit{Int}\ \{N \geqslant 0\}\\ \textbf{var}\ \ x, y : \mathit{Int}\\ x, y := 0, 1\\ \textbf{do}\ x \neq N \rightarrow x, y := x + 1, y + y\ \textbf{od}\\ \{y = 2^N\}\end{array}$$

Solution: Denote $y = 2^x \wedge x \leqslant N$ by P. Use P as the invariant and $N - x$ as bound.

$$\begin{array}{ll}\textbf{con}\ N : \mathit{Int}\ \{N \geqslant 0\} & \\ \textbf{var}\ \ x, y : \mathit{Int} & \\ x, y := 0, 1 & \text{-- Pf0}\\ \{P, \mathit{bnd} : N - x\} & \text{-- Pf2}\\ \textbf{do}\ x \neq N \rightarrow \{P \wedge x \neq N\}\ x, y := x + 1, y + y\ \{P\}\ \textbf{od} & \text{-- Pf1}\\ \{y = 2^N\} & \text{-- Pf3}\end{array}$$

Pf0.

$$\begin{array}{ll} & (y = 2^x\ \wedge\ x \leqslant N)[x, y \backslash 0, 1]\\ \equiv & 1 = 2^0\ \wedge\ 0 \leqslant N\\ \Leftarrow & 0 \leqslant N\ .\end{array}$$

Pf1.

$$\begin{array}{ll} & (y = 2^x\ \wedge\ x \leqslant N)[x, y \backslash x + 1, y + y]\\ \equiv & y + y = 2^x\ \wedge\ x + 1 \leqslant N\\ \Leftarrow & y = 2^x\ \wedge\ x \leqslant N \wedge x \neq N.\end{array}$$

Pf2. It is certainly true that

$$y = 2^x\ \wedge\ x \leqslant N \wedge x \neq N \quad \Rightarrow \quad N - x \geqslant 0.$$

(Note that this is why we need $x \leqslant N$ in the invariant.) Furthermore,

$$\begin{array}{ll} & (N - x < C)[x, y \backslash x + 1, y + y]\\ \equiv & N - x - 1 < C\\ \Leftarrow & N - x = C\\ \Leftarrow & y = 2^x\ \wedge\ x \leqslant N \wedge x \neq N \wedge N - x = C.\end{array}$$

Pf3. It is immediate that

$$y = 2^x \;\wedge\; x \leqslant N \wedge x = N \;\;\Rightarrow\;\; y = 2^N.$$

2. Prove the correctness of the following program:

   con $A, B : Int$ $\{A \geqslant 0\}$
   var $r, n : Int$
   $r, a := 0, 0$
   do $a \neq A \rightarrow r, a := r + B, a + 1$ **od**
   $\{r = A \times B\}$

Solution: Denote $r = a \times B \wedge a \leqslant A$ by P. The annotated program is:

con $A, B : Int$ $\{A \geqslant 0\}$
var $r, a : Int$
$r, a := 0, 0$ -- Pf0
$\{r = a \times B \wedge a \leqslant A, bnd : A - a\}$ -- Pf2
do $a \neq A \rightarrow \{P \wedge a \neq A\}\; r, a := r + B, a + 1\; \{P\}$ **od** -- Pf1
$\{r = A \times B\}$ -- Pf3

Pf0.

$$\begin{aligned} & (r = a \times B \wedge a \leqslant A)[r, a \backslash 0, 0] \\ = \;& 0 = 0 \times B \wedge 0 \leqslant A \\ \Leftarrow \;& 0 \leqslant A \;\; . \end{aligned}$$

Pf1.

$$\begin{aligned} & (r = a \times B \wedge a \leqslant A)[r, a \backslash r + B, a + 1] \\ = \;& r + B = (a + 1) \times B \wedge a + 1 \leqslant A \\ = \;& r + B = a \times B + B \wedge a + 1 \leqslant A \\ \Leftarrow \;& r = a \times B \wedge a \leqslant A \;\; . \end{aligned}$$

Pf2. It is immediate that

$$r = a \times B \wedge a \leqslant A \wedge a \neq A \Rightarrow A - a \geqslant 0 \;\; .$$

(Note that this is why we need $a \leqslant A$ in the invariant.) Furthermore,

$$\begin{aligned} & (A - a < C)[r, a \backslash r + B, a + 1] \\ = \;& A - (a + 1) < C \\ \Leftarrow \;& A - a = C \\ \Leftarrow \;& a \times B \wedge a \leqslant A \wedge a \neq A \wedge A - a = C \;\; . \end{aligned}$$

Pf3. It is immediate that

$$r = a \times B \wedge a \leqslant A \wedge \neg\, (a \neq A) \Rightarrow r = A \times B \;\; .$$

3. Prove the correctness of the following program:

con $N : Int\ \{N \geqslant 0\}$
con A : **array** $[0..N)$ **of** Int
var $n, x : Int$

$x, n := 0, 0$
do $n \neq N \rightarrow x, n := x + A[n], n + 1$ **od**
$\{x = \langle \Sigma i : 0 \leqslant i < N : A[i] \rangle\}$

Solution: Denote $x = \langle \Sigma i : 0 \leqslant i < n : A[i] \rangle \wedge 0 \leqslant n \leqslant N$ by P. The annotated program is:

con $N : Int\ \{N \geqslant 0\}$
con A : **array** $[0..N)$ **of** Int
var $n, x : Int$

$x, n := 0, 0$ -- Pf0
$\{P, bnd : N - n\}$ -- Pf2
do $n \neq N \rightarrow \{P \wedge n \neq N\}\ x, n := x + A[n], n + 1\ \{P\}$ **od** -- Pf1
$\{x = \langle \Sigma i : 0 \leqslant i < N : A[i] \rangle\}$ -- Pf3

The proofs are shown below. Pay attention to range splitting, and where we need $0 \leqslant n$ and $n \leqslant N$ respectively.

Pf0. We reason:

$$\begin{aligned}
& (x = \langle \Sigma i : 0 \leqslant i < n : A[i] \rangle \wedge\wedge\ 0 \leqslant n \leqslant N)[x, n \backslash 0, 0] \\
= {} & 0 = \langle \Sigma i : 0 \leqslant i < 0 : A[i] \rangle \wedge 0 \leqslant 0 \leqslant N \\
= {} & 0 = \langle \Sigma i : False : A[i] \rangle \wedge 0 \leqslant N \\
= {} & 0 = 0 \wedge 0 \leqslant N \\
= {} & 0 \leqslant N \ .
\end{aligned}$$

Pf1. We reason:

$$\begin{aligned}
& (x = \langle \Sigma i : 0 \leqslant i < n : A[i] \rangle \wedge 0 \leqslant n \leqslant N)[x, n \backslash x + A[n], n + 1] \\
= {} & x + A[n] = \langle \Sigma i : 0 \leqslant i < n + 1 : A[i] \rangle \wedge 0 \leqslant n + 1 \leqslant N \\
\Leftarrow {} & \quad \{ \text{ splitting off } i = n \text{ (and assuming } 0 \leqslant n\text{), see below } \} \\
& x + A[n] = \langle \Sigma i : 0 \leqslant i < n : A[i] \rangle + A[n] \wedge 0 \leqslant n \wedge 0 \leqslant n + 1 \leqslant N \\
\Leftarrow {} & x = \langle \Sigma i : 0 \leqslant i < n : A[i] \rangle \wedge 0 \leqslant n \wedge 0 \leqslant n + 1 \leqslant N \\
= {} & x = \langle \Sigma i : 0 \leqslant i < n : A[i] \rangle \wedge 0 \leqslant n \leqslant N \wedge n \neq N \ .
\end{aligned}$$

Note that for the "splitting off $i = n$" step to work, we need $0 \leqslant n$. To see that, we review the calculation on the range:

$$\begin{aligned}
& 0 \leqslant i < n + 1 \\
= {} & 0 \leqslant i \wedge i < n + 1 \\
= {} & 0 \leqslant i \wedge (i < n \vee i = n) \\
= {} & (0 \leqslant i \wedge i < n) \vee (0 \leqslant i \wedge i = n) \\
= {} & (0 \leqslant i < n) \vee i = n \ .
\end{aligned}$$

In the last step we are allowed to refine $0 \leqslant i \wedge i = n$ to $i = n$ only if $0 \leqslant n$. Had it be the case that $0 > n$ instead, $0 \leqslant i \wedge i = n$ would reduce to *False*.

Given the range calculation above, we have that assuming $0 \leqslant n$,

$$
\begin{aligned}
& \langle \Sigma i : 0 \leqslant i < n+1 : A[i] \rangle \\
= \; & \langle \Sigma i : (0 \leqslant i < n) \lor i = n : A[i] \rangle \\
= \; & \quad \{ \text{ range splitting (for disjoint ranges) } \} \\
& \langle \Sigma i : 0 \leqslant i < n : A[i] \rangle + \langle \Sigma i : i = n : A[i] \rangle \\
= \; & \quad \{ \text{ one-point rule } \} \\
& \langle \Sigma i : 0 \leqslant i < n : A[i] \rangle + A\,[n] \;\; .
\end{aligned}
$$

Pf2. We do have that

$$x = \langle \Sigma i : 0 \leqslant i < n : A[i] \rangle \land 0 \leqslant n \leqslant N \Rightarrow N - n \geqslant 0 \;\; .$$

(Note that this is why we need $n \leqslant N$ in the invariant.) Furthermore,

$$
\begin{aligned}
& (N - n < C)[x, n \backslash x + A[n], n+1] \\
= \; & N - (n+1) < C \\
\Leftarrow \; & N - n = C \\
\Leftarrow \; & P \land n \neq N \land N - n = C \;\; .
\end{aligned}
$$

Pf3. It is immediate that

$$
\begin{aligned}
& x = \langle \Sigma i : 0 \leqslant i < n : A[i] \rangle \land 0 \leqslant n \leqslant N \land \neg\, (n \neq N) \\
\Rightarrow \; & x = \langle \Sigma i : 0 \leqslant i < N : A[i] \rangle \;\; .
\end{aligned}
$$

4. Prove the correctness of the following program:

con $N : Int \; \{N \geqslant 0\}$
var $y : Int$
$y := 1$
do $y < N \rightarrow y := y + y$ **od**
$\{y \geqslant N \land \langle \exists k : k \geqslant 0 : y = 2^k \rangle\}$

Solution: We let the invariant be $\langle \exists k : k \geqslant 0 : y = 2^k \rangle$. The annotated program is:

con $N : Int \; \{N \geqslant 0\}$
var $y : Int$
$y := 1$ -- Pf0
$\{\langle \exists k : k \geqslant 0 : y = 2^k \rangle, bnd : N - y\}$ -- Pf1
do $y < N \rightarrow y := y + y$ **od** -- Pf2
$\{y \geqslant N \land \langle \exists k : k \geqslant 0 : y = 2^k \rangle\}$ -- Pf3

Pf_0. We reason:

$$
\begin{aligned}
& \langle \exists k : k \geqslant 0 : y = 2^k \rangle [y \backslash 1] \\
\equiv \; & \langle \exists k : k \geqslant 0 : 1 = 2^k \rangle \\
\Leftarrow \; & \quad \{ \text{ range weakening } \} \\
& \langle \exists k : k = 0 : 1 = 2^k \rangle \\
\equiv \; & \quad \{ \text{ one-point rule } \} \\
& 1 = 2^0 \\
\equiv \; & True \;\; .
\end{aligned}
$$

Pf$_1$. Apparently $y < N$ implies $N - y \geqslant 0$. To prove that the bound decreases, we reason:

$$
\begin{aligned}
& (N - y < C)[y \backslash y + y] \\
\equiv\ & N - (y + y) < C \\
\Leftarrow\ & N - y = C \wedge y > 0 \\
\Leftarrow\ & N - y = C \wedge \langle \exists k : k = 0 : 1 = 2^k \rangle \ .
\end{aligned}
$$

Pf$_2$. We reason:

$$
\begin{aligned}
& \langle \exists k : k \geqslant 0 : y = 2^k \rangle [y \backslash y + y] \\
\equiv\ & \langle \exists k : k \geqslant 0 : y + y = 2^k \rangle \\
\Leftarrow\ & \langle \exists k : k \geqslant 0 : y = 2^k \rangle \ .
\end{aligned}
$$

Pf$_3$. Immediate.

5. Given integers $N \geqslant 0$ and $M > 0$, the following program computes integral division N / M. Prove its correctness.

```
con N, M : Int {N ⩾ 0 ∧ M > 0}
var l, r : Int
l, r := 0, N + 1
do l + 1 ≠ r →
   if ((l + r) / 2) × M ⩽ N → l := (l + r) / 2
   | ((l + r) / 2) × M > N → r := (l + r) / 2
   fi
od
{l × M ⩽ N < (l + 1) × M}
```

Solution: Let $P \equiv l \times M \leqslant N < r \times M \wedge 0 \leqslant l < r$. Use P as the invariant and $r - l$ as bound.

```
con N, M : Int {N ⩾ 0 ∧ M > 0}
var l, r : Int
l, r := 0, N + 1                                   -- Pf0
{l × M ⩽ N < r × M  ∧  0 ⩽ l < r, bnd : r − l}     -- Pf3
do l + 1 ≠ r →
   if ((l + r) / 2) × M ⩽ N → l := (l + r) / 2     -- Pf1
   | ((l + r) / 2) × M > N → r := (l + r) / 2      -- Pf2
   fi
od                                                 -- Pf4
{l × M ⩽ N < (l + 1) × M}
```

Pf$_0$. We reason:

$$
\begin{aligned}
& (l \times M \leqslant N < r \times M \wedge 0 \leqslant l < r)[l, r \backslash 0, N + 1] \\
\equiv\ & 0 \times M \leqslant N < (N + 1) \times M \wedge 0 \leqslant 0 < N + 1 \\
\Leftarrow\ & 0 < M \wedge 0 \leqslant N \ .
\end{aligned}
$$

Pf$_1$. We reason:

$$\begin{aligned}
& (l \times M \leqslant N < r \times M \ \wedge\ 0 \leqslant l < r)[l \backslash (l+r) / 2] \\
\equiv\ & ((l+r) / 2) \times M \leqslant N < r \times M \ \wedge\ 0 \leqslant (l+r) / 2 < r \\
\Leftarrow\ & l \times M \leqslant N < r \times M \ \wedge\ 0 \leqslant l < r \ \wedge \\
& ((l+r) / 2) \times M \leqslant N \ \wedge\ l+1 \neq r \ .
\end{aligned}$$

Pf$_2$. We reason:

$$\begin{aligned}
& (l \times M \leqslant N < r \times M \ \wedge\ 0 \leqslant l < r)[r \backslash (l+r) / 2] \\
\equiv\ & l \times M \leqslant N < ((l+r) / 2) \times M \ \wedge\ 0 \leqslant l < (l+r) / 2 \\
\Leftarrow\ & l \times M \leqslant N < r \times M \ \wedge\ 0 \leqslant l < r \ \wedge \\
& N < ((l+r) / 2) \times M \ \wedge\ l+1 \neq r \ .
\end{aligned}$$

Note that mere $0 \leqslant l < r$ does not guarantee $l < (l+r) / 2$ in integral division. We need $l+1 \neq r$ here.

Pf$_3$. Termination. The invariant guarantees that $r - l \geqslant 0$. We need show that the bound decreases. For the first branch of **if**,

$$\begin{aligned}
& (r - l < C)[l \backslash (l+r) / 2] \\
\equiv\ & r - (l+r) / 2 < C \\
\Leftarrow\ & r - l = C \ \wedge\ l < (l+r) / 2 \\
\equiv\ & \quad \{ \text{integer arithmetic} \} \\
& r - l = C \ \wedge\ 0 \leqslant l < r \ \wedge\ l+1 \neq r \ .
\end{aligned}$$

Note that mere $0 \leqslant l < r$ does not guarantee $l < (l+r) / 2$ in integral division and we do need $l+1 \neq r$ here. For the second branch we reason:

$$\begin{aligned}
& (r - l < C)[r \backslash (l+r) / 2] \\
\equiv\ & ((l+r) / 2) - l < C \\
\Leftarrow\ & r - l = C \ \wedge\ (l+r) / 2 < r \\
\equiv\ & \quad \{ \text{integer arithmetic} \} \\
& r - l = C \ \wedge\ 0 \leqslant l < r \ .
\end{aligned}$$

Pf$_4$. Certainly, $l \times M \leqslant N < r \times M$ and $l + 1 = r$ implies $l \times M \leqslant N < (l+1) \times M$.

6. The following program non-deterministically computes x and y such that $x \times y = N$. Prove:

con $N : Int \ \{N \geqslant 1\}$
var $p, x, y : Int$
$p, x, y := N - 1, 1, 1$
$\{N = x \times y + p \wedge \ldots\}$
do $p \neq 0 \rightarrow$
   if p **mod** $x = 0 \rightarrow y, p := y + 1, p - x$
      | p **mod** $y = 0 \rightarrow x, p := x + 1, p - y$
   fi
od
$\{x \times y = N\}$

Solution: If we try reasoning about the first branch:

$$
\begin{aligned}
& (N = x \times y + p)[y, p \backslash y + 1, p - x] \\
\equiv\ & N = x \times (y + 1) + p - x \\
\equiv\ & N = x \times y + p,
\end{aligned}
$$

we notice that $N = x \times y + p$ does not need the guard $p \textbf{ mod } x$ to hold. The guards, however, do play a role in Pf2 to maintain the invariant.

We use the invariant

$$P\ :\ (N = x \times y + p)\ \wedge\ (0 \leqslant p)\ \wedge\ (0 < x)\ \wedge\ (0 < y)\ \wedge\ (p \textbf{ mod } x = 0 \vee p \textbf{ mod } y = 0)$$

and bound p.

con $N : Int\ \{N \geqslant 1\}$
var $p, x, y : Int$
$p, x, y := N - 1, 1, 1$ -- Pf0
$\{P, bnd : p\}$ -- Pf1
do $p \neq 0 \rightarrow$
  if $p \textbf{ mod } x = 0 \rightarrow \{P \wedge p \neq 0 \wedge p \textbf{ mod } x = 0\}\ y, p := y + 1, p - x\ \{P\}$ -- Pf2
  $|$ $p \textbf{ mod } y = 0 \rightarrow \{P \wedge p \neq 0 \wedge p \textbf{ mod } y = 0\}\ x, p := x + 1, p - y\ \{P\}$ -- Pf3
  fi
  $\{P\}$ -- Pf4
od
$\{x \times y = N\}$ -- Pf5

Pf0.

$$
\begin{aligned}
& P[p, x, y \backslash N - 1, 1, 1] \\
\equiv\ & N = 1 + (N - 1) \wedge 0 \leqslant N - 1 \wedge 0 < 1 \wedge 0 < 1 \wedge ((N - 1) \textbf{ mod } 1 = 0 \vee (N - 1) \textbf{ mod } 1 = 0) \\
\Leftarrow\ & N \geqslant 1\ .
\end{aligned}
$$

Pf1. Apparently $P \wedge \neg(p \neq 0) \Rightarrow p \geqslant 0$. The bound p decreases after the assignment $p := p - x$ because $0 < x$. More precisely, for the first branch:

$$
\begin{aligned}
& (p < C)[y, p \backslash y + 1, p - x] \\
\equiv\ & p - x < C \\
\Leftarrow\ & p = C \wedge x > 0 \\
\Leftarrow\ & p = C \wedge P \wedge p \neq 0.
\end{aligned}
$$

Similarly with the second branch (omitted).

Pf2. We reason:

$$
\begin{aligned}
& (N = x \times y + p\ \wedge\ 0 \leqslant p\ \wedge\ 0 < x\ \wedge\ 0 < y\ \wedge\ (p \textbf{ mod } x = 0 \vee p \textbf{ mod } y = 0))[y, p \backslash y + 1, p - x] \\
\equiv\ & N = x \times (y + 1) + (p - x)\ \wedge\ 0 \leqslant p - x\ \wedge\ 0 < x\ \wedge\ 0 < y + 1\ \wedge \\
& \quad ((p - x) \textbf{ mod } x = 0 \vee (p - x) \textbf{ mod } (y + 1) = 0) \\
\Leftarrow\ & N = x \times y + p\ \wedge\ 0 \leqslant p\ \wedge\ 0 < x\ \wedge\ 0 < y\ \wedge\ (p \textbf{ mod } x = 0 \vee p \textbf{ mod } y = 0) \wedge p \textbf{ mod } x = 0.
\end{aligned}
$$

Examine more closely how the last $\Leftarrow$ holds.

(a) $N = x \times (y+1) + (p - x)$ and $N = x \times y + p$ are equivalent;

(b) $0 \leqslant p - x$ follows from $p \neq 0$ and p **mod** $x = 0$ (if $p < x$, p **mod** x would be p);

(c) $((p - x)$ **mod** $x = 0 \vee (p - x)$ **mod** $(y + 1) = 0)$, being a disjunction, follows from p **mod** $x = 0$.

Pf3. We reason:

$$
\begin{aligned}
& (N = x \times y + p \ \wedge\ 0 \leqslant p \ \wedge\ 0 < x \ \wedge\ 0 < y \ \wedge\ (p \ \mathbf{mod}\ x = 0 \vee p \ \mathbf{mod}\ y = 0))[x, p \backslash x + 1, p - y] \\
\equiv\ & N = (x + 1) \times y + (p - y) \ \wedge\ 0 \leqslant p - y \ \wedge\ 0 < x + 1 \ \wedge\ 0 < y \wedge \\
& \quad ((p - y) \ \mathbf{mod}\ (x + 1) = 0 \vee (p - y) \ \mathbf{mod}\ y = 0) \\
\Leftarrow\ & N = x \times y + p \ \wedge\ 0 \leqslant p \ \wedge\ 0 < x \ \wedge\ 0 < y \ \wedge\ (p \ \mathbf{mod}\ x = 0 \vee p \ \mathbf{mod}\ y = 0) \wedge p \ \mathbf{mod}\ y = 0.
\end{aligned}
$$

Pf4. Here we only have to show that p **mod** $x = 0 \vee p$ **mod** $y = 0$, which is included in the invariant P.

Pf5. Certainly, $P \wedge p = 0 \Rightarrow x \times y = N$.

7. Prove the correctness of the following program:

```
con N : Int {N ⩾ 0}
var x, y : Int
x, y := 0, 0
do x ≠ 0  → x := x − 1
 |  y ≠ N → x, y := x + 1, y + 1
od
{x = 0 ∧ y = N}
```

Solution: Apparently the negation of the guards equivals $x = 0 \wedge y = N$. The difficult part is the proof of termination.

The variable x decreases in one of the branches, therefore we might want to have x in the bound. The variable y increases, therefore we might want $-y$ in the bound. And since each time y increment, x increment too, we weigh y twice as much as x. That gives us $x - 2 \times y$. And since the final value of $x - 2 \times y$ would be $-2\ N$, we add $2\ N$ to the bound. Thus we pick the bound to be $x + 2 \times (N - y)$.

Let the invariant be $P \equiv 0 \leqslant x \wedge 0 \leqslant y \leqslant N$. The annotated program is:

```
con N : Int {N ⩾ 0}
var x, y : Int
x, y := 0, 0                          -- Pf0
{P, bnd : x + 2 × (N − y)}            -- Pf1
do x ≠ 0  → x := x − 1                -- Pf2
 |  y ≠ N → x, y := x + 1, y + 1      -- Pf3
od
{x = 0 ∧ y = N}                       -- Pf4
```

Pf0. We reason:

$$
\begin{aligned}
& P[x, y \backslash 0, 0] \\
\equiv\ & 0 \leqslant 0 \wedge 0 \leqslant 0 \leqslant N \\
\equiv\ & 0 \leqslant N \ .
\end{aligned}
$$

Pf1. It is immediate that $P \wedge (x \neq 0 \vee y \neq N)$ implies $bnd \geqslant 0$. That the first branch decreases the bound is apparent. For the second branch we reason:

$$\begin{aligned} & (x + 2 \times (N - y) < C)[x, y \backslash x + 1, y + 1] \\ \equiv\ & (x + 1) + 2 \times (N - y - 1) < C \\ \equiv\ & x + 2 \times (N - y) + 1 - 2 < C \\ \Leftarrow\ & x + 2 \times (N - y) = C \ . \end{aligned}$$

Pf2.

$$\begin{aligned} & (0 \leqslant x \wedge 0 \leqslant y \leqslant N)[x \backslash x - 1] \\ \equiv\ & 0 \leqslant x - 1 \wedge 0 \leqslant y \leqslant N \\ \equiv\ & 0 \leqslant x \wedge 0 \leqslant y \leqslant N \wedge x \neq 0. \end{aligned}$$

Pf3.

$$\begin{aligned} & (0 \leqslant x \wedge 0 \leqslant y \leqslant N)[x, y \backslash x + 1, y + 1] \\ \equiv\ & 0 \leqslant x + 1 \wedge 0 \leqslant y + 1 \leqslant N \\ \Leftarrow\ & 0 \leqslant x \wedge 0 \leqslant y \leqslant N \wedge y \neq N. \end{aligned}$$

Pf4. Apparently, $\neg(x \neq 0 \vee y \neq N) \equiv x = 0 \wedge y = N$, and thus $P \wedge \neg(x \neq 0 \vee y \neq N) \Rightarrow x = 0 \wedge y = N$.

8. Prove the correctness of the following program:

$$\begin{array}{l} \textbf{con}\ N : Int\ \{N \geqslant 0\} \\ \textbf{var}\ x, y : Int \\ x, y := 0, 0 \\ \textbf{do}\ x \neq 0 \ \ \rightarrow x := x - 1 \\ \mid \ \ y \neq N \rightarrow x, y := N, y + 1 \\ \textbf{od} \\ \{x = 0 \wedge y = N\} \end{array}$$

Solution: Again, the negation of the guards equivals $x = 0 \wedge y = N$ and the difficult part is the proof of termination.

Since x decrements in one of the branches, we might want x in the bound. In another branch, $N - y$ decrements. However, x is set to N each time y decrements by 1. To balance that, one possible guess for the bound is $x + N \times (N - y)$. This turns out to be not sufficient (see Pf_1 below) — we need the increment of y to decrease the bound a bit more. The bound we choose turns out to be:

$$x + (N + 1) \times (N - y) \ .$$

To prove the bound we use the following P as the loop invariant:

$$P \ \equiv \ 0 \leqslant x \leqslant N \wedge 0 \leqslant y \leqslant N \ .$$

The invariant is only needed for proof of termination.

$$
\begin{array}{ll}
\textbf{con } N : \mathit{Int}\ \{N \geqslant 0\} & \\
\textbf{var } x, y : \mathit{Int} & \\
x, y := 0, 0 & \text{-- Pf0} \\
\{P, \mathit{bnd} : x + (N+1) \times (N - y)\} & \text{-- Pf1} \\
\textbf{do } x \neq 0 \ \rightarrow x := x - 1 & \text{-- Pf2} \\
\mid \quad y \neq N \rightarrow x, y := N, y + 1 & \text{-- Pf3} \\
\textbf{od} & \\
\{x = 0 \land y = N\} & \text{-- Pf4}
\end{array}
$$

Pf0. We reason:

$$
\begin{array}{ll}
& P[x, y\backslash 0, 0] \\
\equiv & 0 \leqslant 0 \leqslant N \land 0 \leqslant 0 \leqslant N \\
\equiv & 0 \leqslant N \ .
\end{array}
$$

Pf1. It is immediate that $P \land (x \neq 0 \lor y \neq N)$ implies $\mathit{bnd} \geqslant 0$. That the first branch decreases the bound is apparent. For the second branch we reason:

$$
\begin{array}{ll}
& (x + (N+1) \times (N - y) < C)[x, y\backslash N, y + 1] \\
\equiv & N + (N+1) \times (N - y - 1) < C \\
\equiv & N + (N+1) \times (N - y) - (N+1) < C \\
\equiv & (-1) + (N+1) \times (N - y) < C \\
\Leftarrow & x + (N+1) \times (N - y) = C \land 0 \leqslant x \ .
\end{array}
$$

Note that, had we use $x + N \times (N - y)$ as the bound, the proof would not go through:

$$
\begin{array}{ll}
& (x + N \times (N - y) < C)[x, y\backslash N, y + 1] \\
\equiv & N + N \times (N - y - 1) < C \\
\equiv & N + N \times (N - y) - N < C \\
\equiv & N \times (N - y) < C \\
\not\Leftarrow & x + N \times (N - y) = C \land 0 \leqslant x \ \text{ (since } x \text{ could be 0).}
\end{array}
$$

Pf2.

$$
\begin{array}{ll}
& (0 \leqslant x \leqslant N \land 0 \leqslant y \leqslant N)[x\backslash x - 1] \\
\equiv & 0 \leqslant x - 1 \leqslant N \land 0 \leqslant y \leqslant N \\
\equiv & 0 \leqslant x \leqslant N \land 0 \leqslant y \leqslant N \land x \neq 0.
\end{array}
$$

Pf3.

$$
\begin{array}{ll}
& (0 \leqslant x \leqslant N \land 0 \leqslant y \leqslant N)[x, y\backslash N, y + 1] \\
\equiv & 0 \leqslant N \leqslant N \land 0 \leqslant y + 1 \leqslant N \\
\Leftarrow & 0 \leqslant x \leqslant N \land 0 \leqslant y \leqslant N \land y \neq N.
\end{array}
$$

Pf4. Apparently, $\neg(x \neq 0 \lor y \neq N) \equiv x = 0 \land y = N$, and thus $P \land \neg(x \neq 0 \lor y \neq N) \Rightarrow x = 0 \land y = N$.