Programming Languages: Imperative Program Construction 5. Loop Construction I

Shin-Cheng Mu

Autumn. 2021

National Taiwan University and Academia Sinica

Dijkstra: "The only effective way to raise the confidence level of a program significantly is to give a convincing proof of its correctness. But one should not first make the program and then prove its correctness, because then the requirement of providing the proof would only increase the poor programmer's burden. On the contrary: the programmer should ..."
"...[let] correctness proof and program grow hand in hand: with the choice of the structure of the correctness proof one designs a program for which this proof is applicable."

- From such a specification:

  con *declarations*
  {*preconditions*}
  prog
  {*postcondition*}

- we hope to derive *prog*.
- We usually work backwards from the post condition.
- The techniques we are about to learn is mostly about constructing loops and loop invariants.

Taking Conjuncts as Invariants

- When the post condition has the form $P \wedge Q$, one may take one of the conjuncts as the invariant and the other as the guard:
    - $\{P\}$ **do** $\neg Q \rightarrow S$ **od** $\{P \wedge Q\}$.
- In some extreme cases, since $P \equiv true \wedge P$, one may try:
    - $\{True\}$ **do** $\neg P \rightarrow S$ **od** $\{P\}$.
- E.g. to sort four variables:

$$
\begin{array}{l}
\{True\} \\
\textbf{do}\ a > b \rightarrow a, b := b, a \\
\mid \quad b > c \rightarrow b, c := c, b \\
\mid \quad c > d \rightarrow c, d := d, c \\
\textbf{od} \\
\{a \leqslant b \leqslant c \leqslant d\}
\end{array}
$$

- *Why does it terminate?*

- Consider the specficication:

$$
\begin{array}{l}
\textbf{con } A, B : Int\{0 \leqslant A \wedge 0 < B\} \\
\textbf{var } q, r : Int \\
divmod \\
\{q = A \textbf{ div } B \wedge r = A \textbf{ mod } B\} \quad .
\end{array}
$$

- The post condition expands to $A = q \times B + r \ \wedge\ 0 \leqslant r \ \wedge\ r < B$.

But Which Conjunct to Choose?

- $q = A \ \mathbf{div}\ B \ \wedge \ r = A \ \mathbf{mod}\ B$ expands to $A = q \times B + r \ \wedge \ 0 \leqslant r \ \wedge \ r < B$. Denote it by R. It leads to a number of possibilities:
- $\{0 \leqslant r \ \wedge \ r < B\}\ \mathbf{do}\ A \neq q \times B + r \rightarrow S \ \mathbf{od}\ \{R\}$,
- $\{A = q \times B + r \ \wedge \ r < B\}\ \mathbf{do}\ 0 > r \rightarrow S \ \mathbf{od}\ \{R\}$, or
- $\{A = q \times B + r \ \wedge \ 0 \leqslant r\}\ \mathbf{do}\ r \geqslant B \rightarrow S \ \mathbf{od}\ \{R\}$, etc.

Try $A = q \times B + r \wedge 0 \leqslant r$ as the invariant and $\neg(r < B)$ as the guard:

$\{P : A = q \times B + r \wedge 0 \leqslant r\}$
do $B \leqslant r \rightarrow$
  $\{P \wedge B \leqslant r\}$

  $\{P\}$
od
$\{P \wedge r < B\}$

Try $A = q \times B + r \wedge 0 \leqslant r$ as the invariant and $\neg(r < B)$ as the guard:

$q, r := 0, A$
$\{P : A = q \times B + r \wedge 0 \leqslant r\}$
do $B \leqslant r \rightarrow$
  $\{P \wedge B \leqslant r\}$

  $\{P\}$
od
$\{P \wedge r < B\}$

- P is established by $q, r := 0, A$.

Try $A = q \times B + r \wedge 0 \leqslant r$ as the invariant and $\neg(r < B)$ as the guard:

$q, r := 0, A$
$\{P : A = q \times B + r \wedge 0 \leqslant r\}$
do $B \leqslant r \rightarrow$
 $\{P \wedge B \leqslant r\}$

 $\{P\}$
od
$\{P \wedge r < B\}$

- P is established by $q, r := 0, A$.
- Choose r as the bound.

Try $A = q \times B + r \wedge 0 \leqslant r$ as the invariant and $\neg(r < B)$ as the guard:

$q, r := 0, A$
$\{P : A = q \times B + r \wedge 0 \leqslant r\}$
do $B \leqslant r \rightarrow$
  $\{P \wedge B \leqslant r\}$

  $r := r - B$
  $\{P\}$
od
$\{P \wedge r < B\}$

- P is established by $q, r := 0, A$.
- Choose r as the bound.
- Since $B > 0$, try $r := r - B$:

$$
\begin{aligned}
& P[r \backslash r - B] \\
\equiv\; & A = q \times B + r - B \;\wedge\; 0 \leqslant r - B \\
\equiv\; & A = (q - 1) \times B + r \;\wedge\; B \leqslant r.
\end{aligned}
$$

Denote it by P'.

Try $A = q \times B + r \wedge 0 \leqslant r$ as the invariant and $\neg(r < B)$ as the guard:

$$
\begin{array}{l}
q, r := 0, A \\
\{P : A = q \times B + r \wedge 0 \leqslant r\} \\
\textbf{do } B \leqslant r \rightarrow \\
\quad \{P \wedge B \leqslant r\} \\
\\
\quad \{P'\} \\
\quad r := r - B \\
\quad \{P\} \\
\textbf{od} \\
\{P \wedge r < B\}
\end{array}
$$

- P is established by $q, r := 0, A$.
- Choose r as the bound.
- Since $B > 0$, try $r := r - B$:

$$
\begin{aligned}
& P[r \backslash r - B] \\
\equiv\ & A = q \times B + r - B \ \wedge\ 0 \leqslant r - B \\
\equiv\ & A = (q - 1) \times B + r \ \wedge\ B \leqslant r.
\end{aligned}
$$

  Denote it by P'.
-

$$
\begin{aligned}
& P'[q \backslash q + 1] \\
\equiv\ & A = (q + 1 - 1) \times B + r \ \wedge\ B \leqslant r
\end{aligned}
$$

Try $A = q \times B + r \wedge 0 \leqslant r$ as the invariant and $\neg(r < B)$ as the guard:

$$
\begin{array}{l}
q, r := 0, A \\
\{P : A = q \times B + r \wedge 0 \leqslant r\} \\
\textbf{do } B \leqslant r \rightarrow \\
\quad \{P \wedge B \leqslant r\} \\
\quad q := q + 1 \\
\quad \{P'\} \\
\quad r := r - B \\
\quad \{P\} \\
\textbf{od} \\
\{P \wedge r < B\}
\end{array}
$$

- P is established by $q, r := 0, A$.
- Choose r as the bound.
- Since $B > 0$, try $r := r - B$:

$$
\begin{aligned}
& P[r \backslash r - B] \\
\equiv\ & A = q \times B + r - B \ \wedge\ 0 \leqslant r - B \\
\equiv\ & A = (q - 1) \times B + r \ \wedge\ B \leqslant r.
\end{aligned}
$$

  Denote it by P'.

- $$
\begin{aligned}
& P'[q \backslash q + 1] \\
\equiv\ & A = (q + 1 - 1) \times B + r \ \wedge\ B \leqslant r
\end{aligned}
$$

On Constructing Assignments

We will see this pattern often:

- We want to establish:

$$\{x = E \ \wedge \ \ldots\}$$

$$\{x = E \oplus E'\}$$

We will see this pattern often:

- We want to establish:

$$\{x = E \;\wedge\; ...\}$$
$$x := x \oplus E'$$
$$\{x = E \oplus E'\}$$

We will see this pattern often:

- We want to establish:

$$\{x = E \ \wedge \ \dots\}$$
$$x := x \oplus E'$$
$$\{x = E \oplus E'\}$$

- It works because:

$$\begin{aligned} & (x = E \oplus E')[x \backslash x \oplus E'] \\ \equiv \;& x \oplus E' = E \oplus E' \\ \Leftarrow \;& x = E. \end{aligned}$$

- In general, given a function f, to establish:

$$
\begin{aligned}
&\{x = E\} \\
&x := \ldots \\
&\{x = f\,E\}
\end{aligned}
$$

- we can use an assignment $x := f\,x$. It works because

$$
\begin{aligned}
&\quad (x = f\,E)[x \backslash f\,x] \\
&\equiv f\,x = f\,E \\
&\Leftarrow x = E \quad .
\end{aligned}
$$

Replacing Constants by Variables

- Consider the problem:

$$\begin{aligned}&\textbf{con}\ A, B : Int\ \{A \geqslant 0\ \wedge\ B \geqslant 0\}\\ &\textbf{var}\ r : Int\\ &\mathit{exponentiation}\\ &\{r = A^B\}\quad .\end{aligned}$$

- There is not much we can do with a state space consisting of only one variable.
- Replacing constants by variables may yield some possible invariants.
- Again we have several choices: $r = x^B$, $r = A^x$, $r = x^y$, etc.

- Use the invariant $P_0 : r = A^x$, thus $P_0 \ \wedge \ x = B$ implies the post-condition.

$\{r = A^x \qquad\qquad\qquad\qquad \}$

od

$\{r = A^B\}$

- Use the invariant $P_0 : r = A^x$, thus $P_0 \ \wedge \ x = B$ implies the post-condition.

$r, x := 1, 0$
$\{r = A^x \qquad\qquad\qquad\qquad\qquad\qquad\qquad\}$
do $x \neq B \rightarrow$

od
$\{r = A^B\}$

- Use the invariant $P_0 : r = A^x$, thus $P_0 \ \wedge \ x = B$ implies the post-condition.
- Strategy: increment x in the loop. An upper bound $P_1 : x \leqslant B$.

$r, x := 1, 0$
$\{r = A^x \ \wedge \ x \leqslant B, bnd : B - x\}$
do $x \neq B \rightarrow$

  $x := x + 1$
od
$\{r = A^B\}$

Exponentiation

- Use the invariant $P_0 : r = A^x$, thus $P_0 \ \wedge \ x = B$ implies the post-condition.
- Strategy: increment x in the loop. An upper bound $P_1 : x \leqslant B$.
- $(r = A^x)[x \backslash x + 1] \equiv r = A^{x+1}$. However, when $r = A^x$ holds, $A^{x+1} = A \times A^x = A \times r$!

$r, x := 1, 0$
$\{r = A^x \ \wedge \ x \leqslant B, bnd : B - x\}$
do $x \neq B \rightarrow$

$\quad \{r = A^{x+1} \ \wedge \ x + 1 \leqslant B\}$
$\quad x := x + 1$
od
$\{r = A^B\}$

- Use the invariant $P_0 : r = A^x$, thus $P_0 \ \wedge\ x = B$ implies the post-condition.
- Strategy: increment x in the loop. An upper bound $P_1 : x \leqslant B$.
- $(r = A^x)[x\backslash x+1] \equiv r = A^{x+1}$. However, when $r = A^x$ holds, $A^{x+1} = A \times A^x = A \times r$!
- Indeed, $(r = A^{x+1})[r\backslash A \times r]$

$$\equiv\ A \times r = A^{x+1}$$

$$\Leftarrow\ r = A^x.$$

$$
\begin{array}{l}
r, x := 1, 0 \\
\{r = A^x \ \wedge\ x \leqslant B, bnd : B - x\} \\
\textbf{do } x \neq B \rightarrow \\
\quad r := A \times r \\
\quad \{r = A^{x+1} \ \wedge\ x + 1 \leqslant B\} \\
\quad x := x + 1 \\
\textbf{od} \\
\{r = A^B\}
\end{array}
$$

- Another simple exercise.
- We talk about it because we need range splitting.

$$\begin{array}{l}\textbf{con } N : Int \ \{0 \leqslant N\}; \quad f : \textbf{array } [0..N) \textbf{ of } Int \\ \textbf{var } x : Int \\ sum \\ \{x = \langle\, \Sigma i : 0 \leqslant i < N : f[i] \,\rangle\}\end{array}$$

Summing Up an Array

con $N : Int\ \{0 \leqslant N\};\ \ f : \textbf{array}\ [0..N)\ \textbf{of}\ Int;$

$\{P : x = \langle\, \Sigma i : 0 \leqslant i < n : f[i]\,\rangle, bnd : N - n\}$
do $n \neq N \rightarrow \{P \wedge n \neq N\}$ $\{P\}$ **od**
$\{x = \langle\, \Sigma i : 0 \leqslant i < N : f[i]\,\rangle\}$

$\textbf{con}\ N : Int\ \{0 \leqslant N\};\ \ f : \textbf{array}\ [0..N)\ \textbf{of}\ Int;$

$n, x := 0, 0$

$\{P : x = \langle\, \Sigma i : 0 \leqslant i < n : f[i]\,\rangle, bnd : N - n\}$

$\textbf{do}\ n \neq N \rightarrow \{P \wedge n \neq N\} \qquad\qquad \{P\}\ \textbf{od}$

$\{x = \langle\, \Sigma i : 0 \leqslant i < N : f[i]\,\rangle\}$

$\mathbf{con}\ N : \mathit{Int}\ \{0 \leqslant N\};\ \ f : \mathbf{array}\ [0..N)\ \mathbf{of}\ \mathit{Int};$

$n, x := 0, 0$
$\{P : x = \langle\, \Sigma i : 0 \leqslant i < n : f[i]\,\rangle, bnd : N - n\}$
$\mathbf{do}\ n \neq N \rightarrow \{P \wedge n \neq N\} \qquad n := n + 1\ \ \{P\}\ \mathbf{od}$
$\{x = \langle\, \Sigma i : 0 \leqslant i < N : f[i]\,\rangle\}$

- Use $N - n$ as bound, try incrementing n:

$$\begin{aligned} & (x = \langle\, \Sigma i : 0 \leqslant i < n : f[i]\,\rangle)[n \backslash n + 1] \\ \equiv\quad & x = \langle\, \Sigma i : 0 \leqslant i < n + 1 : f[i]\,\rangle \\ \\ \equiv\quad & x = \langle\, \Sigma i : 0 \leqslant i < n : f[i]\,\rangle + f[n] \qquad . \end{aligned}$$

$\textbf{con}\ N : Int\ \{0 \leqslant N\};\ \ f : \textbf{array}\ [0..N)\ \textbf{of}\ Int;$

$n, x := 0, 0$

$\{P : x = \langle\, \Sigma i : 0{\leqslant}i{<}n : f[i]\,\rangle\ \wedge\ 0 \leqslant n, bnd : N - n\}$

$\textbf{do}\ n \neq N \rightarrow \{P \wedge n \neq N\} \qquad\qquad n := n + 1\ \ \{P\}\ \textbf{od}$

$\{x = \langle\, \Sigma i : 0{\leqslant}i{<}N : f[i]\,\rangle\}$

- Use $N - n$ as bound, try incrementing n:

$$
\begin{aligned}
& (x = \langle\, \Sigma i : 0{\leqslant}i{<}n : f[i]\,\rangle\ \wedge\ 0 \leqslant n)[n \backslash n + 1] \\
\equiv\quad & x = \langle\, \Sigma i : 0{\leqslant}i{<}n+1 : f[i]\,\rangle\ \wedge\ 0 \leqslant n + 1 \\
\Leftarrow\quad & x = \langle\, \Sigma i : 0{\leqslant}i{<}n+1 : f[i]\,\rangle\ \wedge\ 0 \leqslant n \\
\equiv\quad & x = \langle\, \Sigma i : 0{\leqslant}i{<}n : f[i]\,\rangle + f[n]\ \wedge\ 0 \leqslant n.
\end{aligned}
$$

$\textbf{con}\ N : Int\ \{0 \leqslant N\};\ \ f : \textbf{array}\ [0..N)\ \textbf{of}\ Int;$

$n, x := 0, 0$
$\{P : x = \langle\, \Sigma i : 0 \leqslant i < n : f[i]\,\rangle \ \wedge\ 0 \leqslant n, bnd : N - n\}$
$\textbf{do}\ n \neq N \rightarrow \{P \wedge n \neq N\} \qquad n := n + 1\ \ \{P\}\ \textbf{od}$
$\{x = \langle\, \Sigma i : 0 \leqslant i < N : f[i]\,\rangle\}$

-

$$(x = \langle\, \Sigma i : 0 \leqslant i < n : f[i]\,\rangle + f[n]\ \wedge\ 0 \leqslant n)$$

$$\Leftarrow \quad x = \langle\, \Sigma i : 0 \leqslant i < n : f[i]\,\rangle\ \wedge\ 0 \leqslant n.$$

$\textbf{con}\ N : Int\ \{0 \leqslant N\};\ \ f : \textbf{array}\ [0..N)\ \textbf{of}\ Int;$

$$
\begin{array}{l}
n, x := 0, 0 \\
\{P : x = \langle\, \Sigma i : 0 \leqslant i < n : f[i]\,\rangle\ \wedge\ 0 \leqslant n, bnd : N - n\} \\
\textbf{do}\ n \neq N \rightarrow \{P \wedge n \neq N\}\ \ x := x + f[n];\ \ n := n + 1\ \ \{P\}\ \textbf{od} \\
\{x = \langle\, \Sigma i : 0 \leqslant i < N : f[i]\,\rangle\}
\end{array}
$$

-

$$
\begin{array}{ll}
 & (x = \langle\, \Sigma i : 0 \leqslant i < n : f[i]\,\rangle + f[n]\ \wedge\ 0 \leqslant n)[x \backslash x + f[n]] \\
\equiv & x + f[n] = \langle\, \Sigma i : 0 \leqslant i < n : f[i]\,\rangle + f[n]\ \wedge\ 0 \leqslant n \\
\Leftarrow & x = \langle\, \Sigma i : 0 \leqslant i < n : f[i]\,\rangle\ \wedge\ 0 \leqslant n.
\end{array}
$$