Iteration checklist and Termination

Suppose we have a command with this structure

$\{ \mathcal{P} \}$
$\mathcal{S}$
$\{ \mathcal{I} \}$
while $\mathcal{G}$ do
    $\mathcal{T}$
end while
$\{ \mathcal{R} \}$

Checklist

1. Loop initialization establishes the invariant:
   $$\{\mathcal{P}\}\ \mathcal{S}\ \{\mathcal{I}\} \text{ is correct}$$
2. Termination ensures the postcondition:
   $$\mathcal{I} \wedge \neg\mathcal{G} \Rightarrow \mathcal{R} \text{ is universally true}$$
3. The invariant is preserved:
   $$\{\mathcal{I} \wedge \mathcal{G}\}\ \mathcal{T}\ \{\mathcal{I}\} \text{ is correct}$$
4. Each iteration brings the state “closer to” $\neg\mathcal{G}$

The first three items ensure (partial) correctness.

The last item ensures that the loop terminates. Let’s examine that more closely.

Variants

Usually the way to ensure termination is to find an integer expression $\mathcal{E}$

- that can not decrease below 0

  $\mathcal{I} \Rightarrow \mathcal{E} \geq 0$ is universally true

- that decreases with each iteration of the loop

Example 1 *In the binary search example, a suitable variant is* $r - p$*. We know that*

$$r - p \geq 0$$

because the invariant says

$$0 \leq p \leq r \leq x.\text{length}$$

And since $p < q < r$*, we know that the next value of* $r - p$*, which is either* $r - q$ *or* $q - p$*, will be smaller than* $r - p$*.*

As a general scheme we can write

$\{\ \mathcal{I}\ \}$
while $\mathcal{G}$ do
  $\{\ \mathcal{I} \wedge \mathcal{G}\ \}$
  val $\mathcal{V}$: int := $\mathcal{E}$
  $\{\ \mathcal{V} = \mathcal{E} \wedge \mathcal{I} \wedge \mathcal{G}\ \}$
  ?b
  $\{\ \mathcal{E} < \mathcal{V} \wedge \mathcal{I}\}$
end while

where $\mathcal{E}$ is the variant expression and $\mathcal{V}$ is some fresh variable.

If such an outline is correct, the loop must terminate.

Finding invariants

You can often find an invariant by modifying the loop's postcondition.

We will look at two techniques for finding an invariant based on a postcondition

- Deleting a conjunct
- Replacing an expression by a variable

Guiding us is the desire that the invariant be easy to establish initially.

Deleting a conjunct

Suppose the postcondition $\mathcal{R}$ can be split into two conjuncts $\mathcal{R}_0$ and $\mathcal{R}_1$ so that

$$(\mathcal{R}_0 \wedge \mathcal{R}_1) = \mathcal{R}$$

is universally true or even just

$$(\mathcal{R}_0 \wedge \mathcal{R}_1) \Rightarrow \mathcal{R}$$

is universally true.

We could use one conjunct as an invariant and the other as the stopping condition

$\{\mathcal{R}_0\}$
while $\neg\mathcal{R}_1$ do
  given $\mathcal{R}_0$ and $\neg\mathcal{R}_1$ ensure $\mathcal{R}_0$ while decreasing the variant
end while
$\{\mathcal{R}_0 \wedge \mathcal{R}_1\}$

Example: Designing a divider circuit

We want to divide integer x by integer y.

The precondition is $y > 0 \wedge x \geq 0$.

The postcondition is

$$q \times y \leq x < (q+1) \times y$$

Rewrite the postcondition to make the 'and' explicit.

$$\mathcal{R} : (q \times y \leq x) \wedge (x < (q+1) \times y)$$

Take the first part for the invariant and the negation of the second for the guard

{ $y > 0 \wedge x \geq 0$ }
make q so that $q \times y \leq x$
{ $\mathcal{I} : q \times y \leq x$ }
// variant is $x - q \times y$
while $x \geq (q+1) \times y$ do
  given $x \geq (q+1) \times y$ and $\mathcal{I}$ change q to ensure
  $\mathcal{I}$,decreasing $x - q \times y$
end do
{ $q \times y \leq x \wedge x < (q+1) \times y$ }

Which could be (inefficiently) implemented by

{ $y > 0 \wedge x \geq 0$ }
$q := 0$
{ $\mathcal{I} : q \times y \leq x$ } // variant is $x - q \times y$
while $x \geq (q+1) \times y$ do
  $q := q + 1$
end do
{ $q \times y \leq x \wedge x < (q+1) \times y$ }

Replace an expression by a variable

Suppose $\mathcal{R}$ is a postcondition and we can find a condition $\mathcal{I}$ so that $\mathcal{I}[\mathcal{V} : \mathcal{E}]$ implies $\mathcal{R}$, for some variable $\mathcal{V}$ and expression E.

We can take $\mathcal{I}$ to be the invariant and $\mathcal{V} \neq \mathcal{E}$ to be the guard

{ $\mathcal{P}$ }
initialize $\mathcal{V}$ so that $\mathcal{I}$
{ $\mathcal{I}$ }
while $\mathcal{V} \neq \mathcal{E}$ do
    ?given $\mathcal{I}$ and $\mathcal{V} \neq \mathcal{E}$ ensure $\mathcal{I}$, while decreasing a variant
end while
{ $\mathcal{R}$ }

Example: An abstract binary search.

Notation $\{m, .., n\}$ is the set of integers from m up to *and including* n.

$$\{m, .., n\} = \{i \in \mathbb{Z} \mid m \le i \le n\}$$

Let m and n be integers with $m < n$.

Let A be a boolean function defined on $\{m, .., n\}$ such that $\neg A(m)$ and $A(n)$.

Problem: Find an “up edge”, i.e., a point p such that $\neg A(p) \wedge A(p+1)$.

Precondition:

$$\neg A(m) \wedge A(n) \wedge m < n$$

Postcondition: p is an up edge

$$\mathcal{R} : \neg A(p) \wedge A(p+1)$$

By **replacing the expression** $p+1$ **with a variable**, we get a candidate invariant

$$\mathcal{J} : \neg A(p) \wedge A(r)$$

Note that $\mathcal{J}[r : p+1]$ is $\mathcal{R}$.

For a variant, we'll use $r - p$. Now we have a skeleton:

{ $\neg A(m) \wedge A(n) \wedge m < n$ }
$p := m \qquad r := n$
{ $\mathcal{J}$ }
// variant is $r - p$
while $r \neq p + 1$ do
    ?given $\mathcal{J}$ and $r \neq p + 1$ establish $\mathcal{J}$, decreasing $r - p$
end while
{ $\mathcal{R} : \neg A(p) \wedge A(p+1)$ }

But wait! how do we know that $r - p$ is nonnegative. Also how do we know that $A(p)$ and $A(r)$ are well defined. We need a stronger invariant.

It is tempting to conjoin

$$m \le p \le r \le n$$

to $\mathcal{J}$.

However, we also need an invariant that, together with $r \neq p + 1$, ensures $r - p$ is at least 1 (so it can be decreased without violating the invariant). $p \le r$ will not do that (consider $p = r$), but $p < r$ will.

This leads us to an invariant

$$\mathcal{I} : m \le p < r \le n \wedge \neg A(p) \wedge A(r)$$

Note that $\mathcal{I}[r : p + 1]$ implies $\mathcal{R}$.

Revising the skeleton we get

$\{ \neg A(m) \wedge A(n) \wedge m < n \}$
$p := m$
$r := n$
$\{ \mathcal{I} : m \le p < r \le n \wedge \neg A(p) \wedge A(r) \}$
// variant is $r - p$
while $r \neq p + 1$ do
    given $\mathcal{I}$ and $r \neq p + 1$ establish $\mathcal{I}$,decreasing $r - p$
end while
$\{ \neg A(p) \wedge A(p + 1) \}$

Now $p < r$ together with $r \neq p+1$ ensures that $r - p$ is at least 2. Thus the (integer part of the) average of p and r will be such that

$$p < \left\lfloor \frac{p+r}{2} \right\rfloor < r$$

Thus we can implement the loop body

given $\mathcal{I}$ and $r \neq p+1$, establish $\mathcal{I}$ decreasing $r - p$

by

$q := \left\lfloor \frac{p+r}{2} \right\rfloor$
$\{\ p < q < r \wedge \mathcal{I}\ \}$
if $A(q)$ then $r := q$ else $p := q$ end if

In summary, we have

$\{\ \neg A(m) \wedge A(n) \wedge m < n\ \}$
$p := m$
$r := n$
$\{\ \mathcal{I} : m \leq p < r \leq n \wedge \neg A(p) \wedge A(r)\ \}$
// variant is $p - r$
while $r \neq p+1$ do
  $\{\ \mathcal{I} \wedge r \neq p+1\ \}$
  $q := \left\lfloor \frac{p+r}{2} \right\rfloor$
  $\{\ p < q < r \wedge \mathcal{I}\ \}$
  if $A(q)$ then $r := q$ else $p := q$ end if
end while
$\{\ \mathcal{R} : \neg A(p) \wedge A(p+1)\ \}$