Program Correctness

Block 3

Jorge A. Pérez
(based on slides by Arnold Meijster)

Bernoulli Institute for Mathematics, Computer Science, and AI
University of Groningen, Groningen, the Netherlands

February 16, 2023

Outline

From Last Lecture

- Euclid's algorithm (gcd)

Initialization and Active Finalization

- Exercise 6.5
- Exercise 6.7

How to find a good invariant?

- Heuristic: Split conjuncts
- Heuristic: Replace constant by variable
- Heuristic: Generalization

Examples

Stepwise design of a while program

The starting point is the specification $\{P\}\ T\ \{Q\}$.

0 Based on the specification, we decide that we need a loop.

1 Choose an invariant J and a guard B such that

$$J \wedge \neg B \Rightarrow Q \qquad \text{(aka finalization)}$$

2 Initialization: Find a command T_0 such that

$$\{P\}\ T_0\ \{J\}$$

3 Variant function: Choose a $vf \in \mathbb{Z}$ and prove

$$J \wedge B \Rightarrow vf \geq 0$$

4 Body of the loop: Find a command S such that

$$\{J \wedge B \wedge vf = V\}\ S\ \{J \wedge vf < V\}$$

5 Conclude that

$$\{P\}\ T_0;\ \textbf{while}\ B\ \textbf{do}\ S\ \textbf{end}\ \{Q\}$$

Example: Euclid's algorithm (gcd)

We consider the following specification for computing the greatest common divisor of x and y, denoted $\mathtt{gcd}(x, y)$:

var $x,\ y :\ \mathbb{Z}$;
  $\{P:\ x > 0 \wedge y > 0 \wedge \mathtt{gcd}(x, y) = Z\}$
S
  $\{Q:\ x = Z\}$

Example: Euclid's algorithm (gcd)

Before we derive an algorithm, recall that if $x, y \in \mathbb{Z}$ and $y > 0$, then x **div** y and x **mod** y are integers that satisfy

$$(x = y \cdot (x \textbf{ div } y) + x \textbf{ mod } y) \wedge 0 \leq x \textbf{ mod } y < y$$

Example: Euclid's algorithm (gcd)

Before we derive an algorithm, recall that if $x, y \in \mathbb{Z}$ and $y > 0$, then x **div** y and x **mod** y are integers that satisfy

$$(x = y \cdot (x \textbf{ div } y) + x \textbf{ mod } y) \wedge 0 \leq x \textbf{ mod } y < y$$

Also, if z divides y, then (z divides $i \cdot y + j$) $\equiv$ (z divides j).

Example: Euclid's algorithm (gcd)

Before we derive an algorithm, recall that if $x, y \in \mathbb{Z}$ and $y > 0$, then x **div** y and x **mod** y are integers that satisfy

$$(x = y \cdot (x \textbf{ div } y) + x \textbf{ mod } y) \land 0 \leq x \textbf{ mod } y < y$$

Also, if z divides y, then (z divides $i \cdot y + j$) $\equiv$ (z divides j).

Using these facts, we can prove that
- every common divisor of x and $y > 0$
is also
- a common divisor of y and x **mod** y (and vice versa).

Example: Euclid's algorithm (gcd)

Before we derive an algorithm, recall that if $x, y \in \mathbb{Z}$ and $y > 0$, then x **div** y and x **mod** y are integers that satisfy

$$(x = y \cdot (x \textbf{ div } y) + x \textbf{ mod } y) \wedge 0 \leq x \textbf{ mod } y < y$$

Also, if z divides y, then (z divides $i \cdot y + j$) $\equiv$ (z divides j).

Using these facts, we can prove that
- every common divisor of x and $y > 0$
is also
- a common divisor of y and x **mod** y (and vice versa).

We can therefore use the recurrence:

$$\begin{array}{lcl} x > 0 & \Rightarrow & \gcd(x, 0) = x \\ y > 0 & \Rightarrow & \gcd(x, y) = \gcd(y, x \textbf{ mod } y) \end{array}$$

Example: Designing a loop (1/4)

$\{P: \; x > 0 \wedge y > 0 \wedge \mathrm{gcd}(x, y) = Z\}$

S

$\{Q: \; x = Z\}$

0 We decide that we need a **while**: Using the recurrence we expect to decrease the values of x and y iteratively.

Example: Designing a loop (1/4)

$$\{P:\ x>0 \wedge y>0 \wedge \mathrm{gcd}(x,y)=Z\}$$
$$S$$
$$\{Q:\ x=Z\}$$

0 We decide that we need a **while**: Using the recurrence we expect to decrease the values of x and y iteratively.

1 Choose an invariant J and a guard B such that $J \wedge \neg B \Rightarrow Q$.

$$J : x > 0 \wedge y \geq 0 \wedge \mathrm{gcd}(x,y) = Z$$
$$B : y \neq 0$$

Notice:

$$\begin{aligned} & J \wedge \neg B \equiv x > 0 \wedge y \geq 0 \wedge \mathrm{gcd}(x,y) = Z \wedge y = 0 \\ & \quad \{\text{logic; substitution } y = 0\} \\ & \Rightarrow x > 0 \wedge \mathrm{gcd}(x,0) = Z \\ & \quad \{\ \mathrm{gcd}(x,0) = x\ \} \\ & \Rightarrow Q:\ x = Z \end{aligned}$$

Example: Designing a loop (2/4)

Up to here:

$$\boxed{\begin{aligned} P &: x > 0 \wedge y > 0 \wedge \gcd(x, y) = Z \\ J &: x > 0 \wedge y \geq 0 \wedge \gcd(x, y) = Z \\ B &: y \neq 0 \end{aligned}}$$

2 Initialization: Find a command T_0 such that $\{P\}\ T_0\ \{J\}$.
Because $y > 0 \Rightarrow y \geq 0$, we have $P \Rightarrow J$.
Therefore, initialization is not necessary, and $T_0 =$ **skip**.

3 Variant function: Choose a $vf \in \mathbb{Z}$ and prove $J \wedge B \Rightarrow vf \geq 0$
Since J ensures $y \geq 0$, we can simply choose $vf = y$.
Clearly, we have: $J \wedge B \Rightarrow J \Rightarrow y \geq 0 \equiv vf \geq 0$.

Example: Designing a loop (3/4)

4 Body: Find S such that $\{J \wedge B \wedge vf = V\}\ S\ \{J \wedge vf < V\}$

$\{J \wedge B \wedge vf = V\}$

$\{J \wedge vf < V\}$

Example: Designing a loop (3/4)

4 Body: Find S such that $\{J \wedge B \wedge vf = V\}$ S $\{J \wedge vf < V\}$

$\{J \wedge B \wedge vf = V\}$

(* *definitions* J, B, *and* vf *)

$\{\underbrace{x > 0 \wedge y \geq 0 \wedge \mathrm{gcd}(x, y) = Z}_{J} \wedge y \neq 0 \wedge y = V\}$

$\{J \wedge vf < V\}$

Example: Designing a loop (3/4)

4 Body: Find S such that $\{J \wedge B \wedge vf = V\}\ S\ \{J \wedge vf < V\}$

$\{J \wedge B \wedge vf = V\}$

(* *definitions* J, B, *and* vf *)

$\{\underbrace{x > 0 \wedge y \geq 0 \wedge \gcd(x, y) = Z}_{J} \wedge y \neq 0 \wedge y = V\}$

(* *recurrence;* $y > 0 \Rightarrow \gcd(x, y) = \gcd(y, x\ \textbf{mod}\ y)$ *)

$\{J \wedge vf < V\}$

Example: Designing a loop (3/4)

4 Body: Find S such that $\{J \wedge B \wedge vf = V\}$ S $\{J \wedge vf < V\}$

$$\{J \wedge B \wedge vf = V\}$$

(* *definitions* J, B, *and* vf *)

$$\{\underbrace{x > 0 \wedge y \geq 0 \wedge \gcd(x, y) = Z}_{J} \;\wedge\; y \neq 0 \wedge y = V\}$$

(* *recurrence;* $y > 0 \Rightarrow \gcd(x, y) = \gcd(y, x \textbf{ mod } y)$ *)

$$\{y > 0 \wedge \gcd(y, x \textbf{ mod } y) = Z \wedge 0 \leq x \textbf{ mod } y < y = V\}$$

$$\{J \wedge vf < V\}$$

Example: Designing a loop (3/4)

4 Body: Find S such that $\{J \wedge B \wedge vf = V\}\ S\ \{J \wedge vf < V\}$

$\{J \wedge B \wedge vf = V\}$

(* *definitions* J, B, *and* vf *)

$\{\underbrace{x > 0 \wedge y \geq 0 \wedge \gcd(x, y) = Z}_{J} \ \wedge\ y \neq 0 \wedge y = V\}$

(* *recurrence;* $y > 0 \Rightarrow \gcd(x, y) = \gcd(y, x \textbf{ mod } y)$ *)

$\{y > 0 \wedge \gcd(y, x \textbf{ mod } y) = Z \wedge 0 \leq x \textbf{ mod } y < y = V\}$

$m := x$ **mod** y;

$\{y > 0 \wedge \gcd(y, m) = Z \wedge 0 \leq m < y = V\}$

$\{J \wedge vf < V\}$

Example: Designing a loop (3/4)

4 Body: Find S such that $\{J \wedge B \wedge vf = V\}\ S\ \{J \wedge vf < V\}$

$\{J \wedge B \wedge vf = V\}$

(* *definitions* J, B, *and* vf *)

$\{\underbrace{x > 0 \wedge y \geq 0 \wedge \gcd(x,y) = Z}_{J} \wedge y \neq 0 \wedge y = V\}$

(* *recurrence;* $y > 0 \Rightarrow \gcd(x,y) = \gcd(y, x \textbf{ mod } y)$ *)

$\{y > 0 \wedge \gcd(y, x \textbf{ mod } y) = Z \wedge 0 \leq x \textbf{ mod } y < y = V\}$

$m := x$ **mod** y;

$\{y > 0 \wedge \gcd(y, m) = Z \wedge 0 \leq m < y = V\}$

(* *logic* *)

$\{y > 0 \wedge \gcd(y, m) = Z \wedge m \geq 0 \wedge m < V\}$

$\{J \wedge vf < V\}$

Example: Designing a loop (3/4)

4 Body: Find S such that $\{J \wedge B \wedge vf = V\}\ S\ \{J \wedge vf < V\}$

$\{J \wedge B \wedge vf = V\}$

(* *definitions* J, B, *and* vf *)

$\{\underbrace{x > 0 \wedge y \geq 0 \wedge \gcd(x, y) = Z}_{J} \ \wedge\ y \neq 0 \wedge y = V\}$

(* *recurrence;* $y > 0 \Rightarrow \gcd(x, y) = \gcd(y, x \textbf{ mod } y)$ *)

$\{y > 0 \wedge \gcd(y, x \textbf{ mod } y) = Z \wedge 0 \leq x \textbf{ mod } y < y = V\}$

$m := x$ **mod** y;

$\{y > 0 \wedge \gcd(y, m) = Z \wedge 0 \leq m < y = V\}$

(* *logic* *)

$\{y > 0 \wedge \gcd(y, m) = Z \wedge m \geq 0 \wedge m < V\}$

$x := y$;

$\{x > 0 \wedge \gcd(x, m) = Z \wedge m \geq 0 \wedge m < V\}$

$\{J \wedge vf < V\}$

Example: Designing a loop (3/4)

4 Body: Find S such that $\{J \wedge B \wedge vf = V\}\ S\ \{J \wedge vf < V\}$

$\{J \wedge B \wedge vf = V\}$

(* *definitions* J, B, *and* vf *)

$\{\underbrace{x > 0 \wedge y \geq 0 \wedge \gcd(x, y) = Z}_{J} \ \wedge\ y \neq 0 \wedge y = V\}$

(* *recurrence;* $y > 0 \Rightarrow \gcd(x, y) = \gcd(y, x \textbf{ mod } y)$ *)

$\{y > 0 \wedge \gcd(y, x \textbf{ mod } y) = Z \wedge 0 \leq x \textbf{ mod } y < y = V\}$

$m := x$ **mod** y;

$\{y > 0 \wedge \gcd(y, m) = Z \wedge 0 \leq m < y = V\}$

(* *logic* *)

$\{y > 0 \wedge \gcd(y, m) = Z \wedge m \geq 0 \wedge m < V\}$

$x := y$;

$\{x > 0 \wedge \gcd(x, m) = Z \wedge m \geq 0 \wedge m < V\}$

$y := m$;

$\{x > 0 \wedge \gcd(x, y) = Z \wedge y \geq 0 \wedge y < V\}$

$\{J \wedge vf < V\}$

Example: Designing a loop (3/4)

4 Body: Find S such that $\{J \wedge B \wedge vf = V\}\ S\ \{J \wedge vf < V\}$

$$\{J \wedge B \wedge vf = V\}$$

(* *definitions* J, B, *and* vf *)

$$\{\underbrace{x > 0 \wedge y \geq 0 \wedge \gcd(x, y) = Z}_{J} \wedge y \neq 0 \wedge y = V\}$$

(* *recurrence;* $y > 0 \Rightarrow \gcd(x, y) = \gcd(y, x \textbf{ mod } y)$ *)

$$\{y > 0 \wedge \gcd(y, x \textbf{ mod } y) = Z \wedge 0 \leq x \textbf{ mod } y < y = V\}$$

$m := x$ **mod** y;

$$\{y > 0 \wedge \gcd(y, m) = Z \wedge 0 \leq m < y = V\}$$

(* *logic* *)

$$\{y > 0 \wedge \gcd(y, m) = Z \wedge m \geq 0 \wedge m < V\}$$

$x := y$;

$$\{x > 0 \wedge \gcd(x, m) = Z \wedge m \geq 0 \wedge m < V\}$$

$y := m$;

$$\{x > 0 \wedge \gcd(x, y) = Z \wedge y \geq 0 \wedge y < V\}$$

(* *definitions* J *and* vf *)

$$\{J \wedge vf < V\}$$

Example: Designing a loop (4/4)

5 We found the following program fragment (Euclid's algorithm):

var $x,\ y,\ m:\ \mathbb{Z}$;
  $\{P:\ x > 0 \land y > 0 \land \mathrm{gcd}(x, y) = Z\}$
  $\{J: x > 0 \land y \geq 0 \land \mathrm{gcd}(x, y) = Z\}$
    (* $vf = y$ *)
while $y \neq 0$ **do**
  $m := x$ **mod** y;
  $x := y$;
  $y := m$;
end;
  $\{Q:\ x = Z\}$

Outline

Proof rule: while-loop

Recall the proof rule for while-loops:

$$\frac{J \wedge B \Rightarrow vf \geq 0 \quad \{J \wedge B \wedge \ vf = V\}\ S\ \{J \wedge \ vf < V\}}{\{J\}\ \textbf{while}\ B\ \ \textbf{do}\ S\ \textbf{end}\ \{J \wedge \neg B\}}$$

Stepwise design of a while program

The starting point is the specification $\{P\}\ T\ \{Q\}$.

Stepwise design of a while program

The starting point is the specification $\{P\}\ T\ \{Q\}$.

0 Based on the specification, we decide that we need a loop.

Stepwise design of a while program

The starting point is the specification $\{P\}\ T\ \{Q\}$.

0 Based on the specification, we decide that we need a loop.
1 Choose an invariant J and a guard B such that

$$J \wedge \neg B \Rightarrow Q \qquad \text{(aka finalization)}$$

Stepwise design of a while program

The starting point is the specification $\{P\}\ T\ \{Q\}$.

0 Based on the specification, we decide that we need a loop.

1 Choose an invariant J and a guard B such that

$$J \wedge \neg B \Rightarrow Q \qquad \text{(aka finalization)}$$

2 Initialization: Find a command T_0 such that

$$\{P\}\ T_0\ \{J\}$$

Stepwise design of a while program

The starting point is the specification $\{P\}\ T\ \{Q\}$.

0 Based on the specification, we decide that we need a loop.

1 Choose an invariant J and a guard B such that

$$J \wedge \neg B \Rightarrow Q \qquad \text{(aka finalization)}$$

2 Initialization: Find a command T_0 such that

$$\{P\}\ T_0\ \{J\}$$

3 Variant function: Choose a $vf \in \mathbb{Z}$ and prove

$$J \wedge B \Rightarrow vf \geq 0$$

Stepwise design of a while program

The starting point is the specification $\{P\}\ T\ \{Q\}$.

0 Based on the specification, we decide that we need a loop.

1 Choose an invariant J and a guard B such that

$$J \wedge \neg B \Rightarrow Q \qquad \text{(aka finalization)}$$

2 Initialization: Find a command T_0 such that

$$\{P\}\ T_0\ \{J\}$$

3 Variant function: Choose a $vf \in \mathbb{Z}$ and prove

$$J \wedge B \Rightarrow vf \geq 0$$

4 Body of the loop: Find a command S such that

$$\{J \wedge B \wedge vf = V\}\ S\ \{J \wedge vf < V\}$$

Stepwise design of a while program

The starting point is the specification $\{P\}\ T\ \{Q\}$.

0 Based on the specification, we decide that we need a loop.

1 Choose an invariant J and a guard B such that

$$J \wedge \neg B \Rightarrow Q \qquad \text{(aka finalization)}$$

2 Initialization: Find a command T_0 such that

$$\{P\}\ T_0\ \{J\}$$

3 Variant function: Choose a $vf \in \mathbb{Z}$ and prove

$$J \wedge B \Rightarrow vf \geq 0$$

4 Body of the loop: Find a command S such that

$$\{J \wedge B \wedge vf = V\}\ S\ \{J \wedge vf < V\}$$

5 Conclude that

$$\{P\}\ T_0;\ \textbf{while}\ B\ \textbf{do}\ S\ \textbf{end}\ \{Q\}$$

Initialization and Active Finalization

(Active) Initialization

- In general, we need a command T_0 to establish the initial validity of the invariant: $\{P\}\ T_0\ \{J\}$.
- If $P \Rightarrow J$ then $T_0 =$ **skip** (e.g. Euclid's algorithm)
- Otherwise, if $P \not\Rightarrow J$ then we need an (active) initialization command T_0.

Initialization and Active Finalization

(Active) Initialization

- In general, we need a command T_0 to establish the initial validity of the invariant: $\{P\}\ T_0\ \{J\}$.
- If $P \Rightarrow J$ then $T_0 =$ **skip** (e.g. Euclid's algorithm)
- Otherwise, if $P \not\Rightarrow J$ then we need an (active) initialization command T_0.

Active Finalization

- Similarly, if $J \wedge \neg B \Rightarrow Q_1$ but $J \wedge \neg B \not\Rightarrow Q$, then we need a command T_1 that establishes the postcondition: $\{Q_1\}\ T_1\ \{Q\}$.
- In this case, we call T_1 an active finalization.

A Generalized Rule

$$\frac{\{P\}\ T_0\ \{J\} \quad \{Q_1\}\ T_1\ \{Q\} \qquad J \wedge B \Rightarrow vf \geq 0 \quad \{J \wedge B \wedge\ vf = V\}\ S\ \{J \wedge\ vf < V\}}{\{P\}\ T_0;\ \{J\}\ \textbf{while}\ B\ \textbf{do}\ S\ \textbf{end};\ \{Q_1\}\ T_1;\ \{Q\}}$$

Specific cases:

- If $T_0 =$ **skip** then initialization is not necessary (and we may need to show that $P \Rightarrow J$).
- If $T_1 =$ **skip** then active finalization is not necessary (and we may need to show that $Q_1 \Rightarrow Q$).

Plan

Rest of Today:

- Exercises 6.5 and 6.7: loops with initialization and finalization.
- Some heuristics for finding a good invariant.

Next week:

- More on recurrence relations (Monday)
- No lecture on Thursday

Exercise 6.5

The function f is defined by the recurrence:

$$\begin{aligned} y \leq 0 &\quad\Rightarrow\quad f(y,z) = z \\ y > 0 &\quad\Rightarrow\quad f(y,z) = 10 \cdot f(y \textbf{ div } 10, z) + y \textbf{ mod } 10 \end{aligned}$$

Find a command S that satisfies the specification:

var $y,\ z :\ \mathbb{Z}$;
$\{P:\ Z = f(y,z)\}$
S
$\{Q:\ Z = z\}$

Use active finalization, auxiliary variables m and n, and

$$J:\ Z = m \cdot f(y,z) + n$$

Exercise 6.5: Initialization

$$
\begin{aligned}
y \le 0 &\Rightarrow f(y,z) = z \\
y > 0 &\Rightarrow f(y,z) = 10 \cdot f(y \textbf{ div } 10, z) + y \textbf{ mod } 10 \\
P &: Z = f(y,z) \\
J &: Z = m \cdot f(y,z) + n \\
Q &: Z = z
\end{aligned}
$$

Because $P \not\Rightarrow J$, we need initialization, but it is easy:

$\{P : Z = f(y,z)\}$
  (* *calculus* *)
$\{Z = 1 \cdot f(y,z) + 0\}$
$m := 1;\ n := 0;$
$\{J :\ Z = m \cdot f(y,z) + n\}$

Exercise 6.5: Guard

$$y \leq 0 \Rightarrow f(y, z) = z$$
$$y > 0 \Rightarrow f(y, z) = 10 \cdot f(y \textbf{ div } 10, z) + y \textbf{ mod } 10$$
$$P : Z = f(y, z)$$
$$J : Z = m \cdot f(y, z) + n$$
$$Q : Z = z$$

We now define the guard B. We know $f(y, z)$ directly if $y \leq 0$. Therefore, we choose B to be $\neg(y \leq 0)$, i.e. $B : y > 0$.

Exercise 6.5: Guard

$$
\begin{aligned}
y \leq 0 &\Rightarrow f(y, z) = z \\
y > 0 &\Rightarrow f(y, z) = 10 \cdot f(y \textbf{ div } 10, z) + y \textbf{ mod } 10 \\
P &: Z = f(y, z) \\
J &: Z = m \cdot f(y, z) + n \\
Q &: Z = z
\end{aligned}
$$

We now define the guard B. We know $f(y, z)$ directly if $y \leq 0$. Therefore, we choose B to be $\neg(y \leq 0)$, i.e. $B : y > 0$.

Given this, $J \wedge \neg B \not\Rightarrow Q$ and so we need active finalization:

$\{J \wedge \neg B\}$
$\{Z = m \cdot f(y, z) + n \wedge y \leq 0\}$
(* *definition f* *)
$\{Z = m \cdot z + n\}$
$z := m * z + n;$
$\{Q : \; Z = z\}$

Exercise 6.5: Variant and Body

Because $B : y > 0$, we need to decrease y until $y \leq 0$.
We choose $vf = y \in \mathbb{Z}$. Clearly, $B \equiv vf > 0$ and $J \wedge B \Rightarrow vf \geq 0$.

Exercise 6.5: Variant and Body

Because $B : y > 0$, we need to decrease y until $y \leq 0$.
We choose $vf = y \in \mathbb{Z}$. Clearly, $B \equiv vf > 0$ and $J \wedge B \Rightarrow vf \geq 0$.

For the body of the loop we find:

$$\{J \wedge B \wedge vf = V\}$$
$$\{Z = m \cdot f(y, z) + n \wedge y > 0 \wedge y = V\}$$

$$\{J \wedge vf < V\}$$

Exercise 6.5: Variant and Body

Because $B : y > 0$, we need to decrease y until $y \leq 0$.
We choose $vf = y \in \mathbb{Z}$. Clearly, $B \equiv vf > 0$ and $J \wedge B \Rightarrow vf \geq 0$.

For the body of the loop we find:

$\{J \wedge B \wedge vf = V\}$
$\{Z = m \cdot f(y, z) + n \wedge y > 0 \wedge y = V\}$
  (* *definition* f *;* $y = V > 0 \Rightarrow y \textbf{ div } 10 < V$ *)
$\{Z = m \cdot (10 \cdot f(y \textbf{ div } 10, z) + y \textbf{ mod } 10) + n \ \wedge \ y \textbf{ div } 10 < V\}$

$\{J \wedge vf < V\}$

Exercise 6.5: Variant and Body

Because $B : y > 0$, we need to decrease y until $y \leq 0$.
We choose $vf = y \in \mathbb{Z}$. Clearly, $B \equiv vf > 0$ and $J \wedge B \Rightarrow vf \geq 0$.

For the body of the loop we find:

$\{J \wedge B \wedge vf = V\}$
$\{Z = m \cdot f(y, z) + n \wedge y > 0 \wedge y = V\}$
  (* *definition* f *;* $y = V > 0 \Rightarrow y \textbf{ div } 10 < V$ *)
$\{Z = m \cdot (10 \cdot f(y \textbf{ div } 10, z) + y \textbf{ mod } 10) + n \ \wedge \ y \textbf{ div } 10 < V\}$
  (* *calculus* *)
$\{Z = 10 \cdot m \cdot f(y \textbf{ div } 10, z) + m \cdot (y \textbf{ mod } 10) + n \ \wedge \ y \textbf{ div } 10 < V\}$

$\{J \wedge vf < V\}$

Exercise 6.5: Variant and Body

Because $B : y > 0$, we need to decrease y until $y \leq 0$.
We choose $vf = y \in \mathbb{Z}$. Clearly, $B \equiv vf > 0$ and $J \wedge B \Rightarrow vf \geq 0$.

For the body of the loop we find:

$\{J \wedge B \wedge vf = V\}$
$\{Z = m \cdot f(y, z) + n \wedge y > 0 \wedge y = V\}$
(* *definition* f *;* $y = V > 0 \Rightarrow y$ **div** $10 < V$ *)
$\{Z = m \cdot (10 \cdot f(y \textbf{ div } 10, z) + y \textbf{ mod } 10) + n \ \wedge \ y \textbf{ div } 10 < V\}$
(* *calculus* *)
$\{Z = 10 \cdot m \cdot f(y \textbf{ div } 10, z) + m \cdot (y \textbf{ mod } 10) + n \ \wedge \ y \textbf{ div } 10 < V\}$
$n := m * (y \textbf{ mod } 10) + n;$
$\{Z = 10 \cdot m \cdot f(y \textbf{ div } 10, z) + n \ \wedge \ y \textbf{ div } 10 < V\}$

$\{J \wedge vf < V\}$

Exercise 6.5: Variant and Body

Because $B : y > 0$, we need to decrease y until $y \leq 0$.
We choose $vf = y \in \mathbb{Z}$. Clearly, $B \equiv vf > 0$ and $J \wedge B \Rightarrow vf \geq 0$.

For the body of the loop we find:

$\{J \wedge B \wedge vf = V\}$
$\{Z = m \cdot f(y, z) + n \wedge y > 0 \wedge y = V\}$
(* *definition f ; $y = V > 0 \Rightarrow y$ **div** $10 < V$* *)
$\{Z = m \cdot (10 \cdot f(y \textbf{ div } 10, z) + y \textbf{ mod } 10) + n \ \wedge \ y \textbf{ div } 10 < V\}$
(* *calculus* *)
$\{Z = 10 \cdot m \cdot f(y \textbf{ div } 10, z) + m \cdot (y \textbf{ mod } 10) + n \ \wedge \ y \textbf{ div } 10 < V\}$
$n := m * (y \textbf{ mod } 10) + n;$
$\{Z = 10 \cdot m \cdot f(y \textbf{ div } 10, z) + n \ \wedge \ y \textbf{ div } 10 < V\}$
$m := 10 * m;$
$\{Z = m \cdot f(y \textbf{ div } 10, z) + n \wedge y \textbf{ div } 10 < V\}$

$\{J \wedge vf < V\}$

Exercise 6.5: Variant and Body

Because $B : y > 0$, we need to decrease y until $y \leq 0$.
We choose $vf = y \in \mathbb{Z}$. Clearly, $B \equiv vf > 0$ and $J \wedge B \Rightarrow vf \geq 0$.

For the body of the loop we find:

$$
\begin{array}{l}
\quad \{J \wedge B \wedge vf = V\} \\
\quad \{Z = m \cdot f(y, z) + n \wedge y > 0 \wedge y = V\} \\
\qquad \text{(* } \textit{definition } f\textit{; } y = V > 0 \Rightarrow y \textbf{ div } 10 < V \text{ *)} \\
\quad \{Z = m \cdot (10 \cdot f(y \textbf{ div } 10, z) + y \textbf{ mod } 10) + n \ \wedge \ y \textbf{ div } 10 < V\} \\
\qquad \text{(* } \textit{calculus} \text{ *)} \\
\quad \{Z = 10 \cdot m \cdot f(y \textbf{ div } 10, z) + m \cdot (y \textbf{ mod } 10) + n \ \wedge \ y \textbf{ div } 10 < V\} \\
n := m * (y \textbf{ mod } 10) + n; \\
\quad \{Z = 10 \cdot m \cdot f(y \textbf{ div } 10, z) + n \ \wedge \ y \textbf{ div } 10 < V\} \\
m := 10 * m; \\
\quad \{Z = m \cdot f(y \textbf{ div } 10, z) + n \wedge y \textbf{ div } 10 < V\} \\
y := y \textbf{ div } 10; \\
\quad \{Z = m \cdot f(y, z) + n \wedge y < V\} \\
\quad \{J \wedge vf < V\}
\end{array}
$$

Exercise 6.5: Conclusion

Using active initialization and finalization, we derived the following program fragment:

$$
\begin{array}{l}
\textbf{var}\ \ n,\ m,\ y,\ z:\ \mathbb{Z};\\
\quad \{P:\ Z = f(y,z)\}\\
m := 1;\\
n := 0;\\
\quad \{J:\ Z = m \cdot f(y,z) + n\}\\
\qquad (^*\ vf = y\ ^*)\\
\textbf{while}\ y > 0\ \textbf{do}\\
\quad n := m * (y\ \textbf{mod}\ 10) + n;\\
\quad m := 10 * m;\\
\quad y := y\ \textbf{div}\ 10;\\
\textbf{end};\\
z := m * z + n;\\
\quad \{Q:\ z = Z\}
\end{array}
$$

Exercise 6.7

The function h is defined by the recurrence:

$$h(0) = 0$$
$$n > 0 \Rightarrow h(n) = 5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4$$

Find a command S that satisfies the following specification:

var $n,\ x:\ \mathbb{Z}$;
$\{P:\ n \geq 0 \land Z = h(n)\}$
S
$\{Q:\ Z = x\}$

Exercise 6.7

The function h is defined by the recurrence:

$$h(0) = 0$$
$$n > 0 \Rightarrow h(n) = 5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4$$

Find a command S that satisfies the following specification:

$$\begin{array}{l} \textbf{var } n,\ x :\ \mathbb{Z}; \\ \quad \{P:\ n \geq 0 \wedge Z = h(n)\} \\ S \\ \quad \{Q:\ Z = x\} \end{array}$$

Introduce a variable y and use the invariant

$$J:\ Z = y \cdot h(n) + x \wedge n \geq 0$$

Exercise 6.7: Initialization

$$\boxed{\begin{aligned} h(0) &= 0 \\ n > 0 \Rightarrow h(n) &= 5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4 \\ P: n \geq 0 \wedge Z &= h(n) \\ J: Z &= y \cdot h(n) + x \wedge n \geq 0 \end{aligned}}$$

Because $P \not\Rightarrow J$, we need initialization,

Exercise 6.7: Initialization

$$
\begin{gathered}
h(0) = 0 \\
n > 0 \Rightarrow h(n) = 5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4 \\
P : n \geq 0 \land Z = h(n) \\
J : Z = y \cdot h(n) + x \land n \geq 0
\end{gathered}
$$

Because $P \not\Rightarrow J$, we need initialization, but it is easy:

$\{P : Z = h(n) \land n \geq 0\}$
  (* *calculus* *)
$\{Z = 1 \cdot h(n) + 0 \land n \geq 0\}$
$x := 0;\ y := 1;$
$\{J : Z = y \cdot h(n) + x \land n \geq 0\}$

Exercise 6.7: Guard and Variant

$$
\begin{aligned}
h(0) &= 0 \\
n > 0 \Rightarrow h(n) &= 5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4 \\
Q &: Z = x \\
J &: Z = y \cdot h(n) + x \wedge n \geq 0
\end{aligned}
$$

We now define the guard B. We know $h(n)$ directly if $n = 0$. Therefore, we choose $B : n \neq 0$.

Exercise 6.7: Guard and Variant

$$
\begin{aligned}
h(0) &= 0 \\
n > 0 \Rightarrow h(n) &= 5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4 \\
Q &: Z = x \\
J &: Z = y \cdot h(n) + x \wedge n \geq 0
\end{aligned}
$$

We now define the guard B. We know $h(n)$ directly if $n = 0$. Therefore, we choose $B : n \neq 0$. We check that $J \wedge \neg B \Rightarrow Q$:

$$
\begin{aligned}
J \wedge \neg B = {} & Z = y \cdot h(n) + x \wedge n \geq 0 \wedge n = 0 \\
& \quad \text{(* } h(0) = 0 \text{ \textit{and logic} *)} \\
\Rightarrow {} & Z = y \cdot 0 + x \\
& \quad \text{(* \textit{calculus} *)} \\
\equiv {} & Z = x
\end{aligned}
$$

Exercise 6.7: Guard and Variant

$$
\begin{array}{c}
h(0) = 0 \\
n > 0 \Rightarrow h(n) = 5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4 \\
Q : Z = x \\
J \; : Z = y \cdot h(n) + x \wedge n \geq 0
\end{array}
$$

We now define the guard B. We know $h(n)$ directly if $n = 0$. Therefore, we choose $B : n \neq 0$. We check that $J \wedge \neg B \Rightarrow Q$:

$$
\begin{aligned}
& J \wedge \neg B = Z = y \cdot h(n) + x \wedge n \geq 0 \wedge n = 0 \\
& \qquad \text{(* } h(0) = 0 \text{ and logic *)} \\
\Rightarrow \; & Z = y \cdot 0 + x \\
& \qquad \text{(* calculus *)} \\
\equiv \; & Z = x
\end{aligned}
$$

As J gives $n \geq 0$ and $B : n \neq 0$, we need to decrease n until $n = 0$. We choose $vf = n \in \mathbb{Z}$. Clearly, $J \wedge B \Rightarrow vf \geq 0$.

Exercise 6.7: Body of the Loop

$\{J \wedge B \wedge vf = V\}$
$\{Z = y \cdot h(n) + x \wedge n \geq 0 \wedge n \neq 0 \wedge n = V\}$

$\{J \wedge vf < V\}$

Exercise 6.7: Body of the Loop

$\{J \wedge B \wedge vf = V\}$
$\{Z = y \cdot h(n) + x \wedge n \geq 0 \wedge n \neq 0 \wedge n = V\}$
  (* $n > 0 \Rightarrow h(n) = 5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4 \wedge 0 \leq n \textbf{ div } 3 < n$ *)
$\{Z = y \cdot (5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$

$\{J \wedge vf < V\}$

Exercise 6.7: Body of the Loop

$\{J \wedge B \wedge vf = V\}$
$\{Z = y \cdot h(n) + x \wedge n \geq 0 \wedge n \neq 0 \wedge n = V\}$
(* $n > 0 \Rightarrow h(n) = 5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4 \wedge 0 \leq n \textbf{ div } 3 < n$ *)
$\{Z = y \cdot (5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$
(* *calculus* *)
$\{Z = 5 \cdot y \cdot h(n \textbf{ div } 3) + y \cdot (n \textbf{ mod } 4) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$

$\{J \wedge vf < V\}$

Exercise 6.7: Body of the Loop

$\{J \wedge B \wedge vf = V\}$

$\{Z = y \cdot h(n) + x \wedge n \geq 0 \wedge n \neq 0 \wedge n = V\}$

(* $n > 0 \Rightarrow h(n) = 5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4 \wedge 0 \leq n \textbf{ div } 3 < n$ *)

$\{Z = y \cdot (5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$

(* *calculus* *)

$\{Z = 5 \cdot y \cdot h(n \textbf{ div } 3) + y \cdot (n \textbf{ mod } 4) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$

$x := y * (n \textbf{ mod } 4) + x;$

$\{Z = 5 \cdot y \cdot h(n \textbf{ div } 3) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$

$\{J \wedge vf < V\}$

Exercise 6.7: Body of the Loop

$\{J \wedge B \wedge vf = V\}$
$\{Z = y \cdot h(n) + x \wedge n \geq 0 \wedge n \neq 0 \wedge n = V\}$
(* $n > 0 \Rightarrow h(n) = 5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4 \wedge 0 \leq n \textbf{ div } 3 < n$ *)
$\{Z = y \cdot (5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$
(* *calculus* *)
$\{Z = 5 \cdot y \cdot h(n \textbf{ div } 3) + y \cdot (n \textbf{ mod } 4) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$
$x := y * (n \textbf{ mod } 4) + x;$
$\{Z = 5 \cdot y \cdot h(n \textbf{ div } 3) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$
$y := 5 * y;$
$\{Z = y \cdot h(n \textbf{ div } 3) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$

$\{J \wedge vf < V\}$

Exercise 6.7: Body of the Loop

$\{J \wedge B \wedge vf = V\}$

$\{Z = y \cdot h(n) + x \wedge n \geq 0 \wedge n \neq 0 \wedge n = V\}$

(* $n > 0 \Rightarrow h(n) = 5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4 \wedge 0 \leq n \textbf{ div } 3 < n$ *)

$\{Z = y \cdot (5 \cdot h(n \textbf{ div } 3) + n \textbf{ mod } 4) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$

(* *calculus* *)

$\{Z = 5 \cdot y \cdot h(n \textbf{ div } 3) + y \cdot (n \textbf{ mod } 4) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$

$x := y * (n \textbf{ mod } 4) + x;$

$\{Z = 5 \cdot y \cdot h(n \textbf{ div } 3) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$

$y := 5 * y;$

$\{Z = y \cdot h(n \textbf{ div } 3) + x \wedge 0 \leq n \textbf{ div } 3 < V\}$

$n := n \textbf{ div } 3;$

$\{Z = y \cdot h(n) + x \wedge 0 \leq n < V\}$

$\{J \wedge vf < V\}$

Exercise 6.7: Conclusion

Using initialization, we derived the following program fragment:

$$
\begin{array}{l}
\textbf{var}\ \ x,\ y,\ n:\ \mathbb{Z};\\
\quad \{P:\ Z = h(n) \wedge n \geq 0\}\\
x := 0;\\
y := 1;\\
\quad \{J:\ Z = y \cdot h(n) + x \wedge n \geq 0\}\\
\qquad (^*\ vf = n\ ^*)\\
\textbf{while}\ n \neq 0\ \textbf{do}\\
\quad x := y * (n\ \textbf{mod}\ 4) + x;\\
\quad y := 5 * y;\\
\quad n := n\ \textbf{div}\ 3;\\
\textbf{end};\\
\quad \{Q:\ x = Z\}
\end{array}
$$

Outline

From Last Lecture
- Euclid's algorithm (gcd)

Initialization and Active Finalization
- Exercise 6.5
- Exercise 6.7

How to find a good invariant?
- Heuristic: Split conjuncts
- Heuristic: Replace constant by variable
- Heuristic: Generalization

Examples

How to find a good invariant?

- Imagine you interrupt the loop, open it up, and take a snapshot:
  - What would you observe? (variables, predicates)
  - What is key to the loop's operation?

How to find a good invariant?

- Imagine you interrupt the loop, open it up, and take a snapshot:
  - What would you observe? (variables, predicates)
  - What is key to the loop's operation?
- Informally, J should be a predicate that is ‘in between’ P and Q.
- Not too weak (uninformative/useless), but
  not too strong (hard to initialize and restore).
- Rule of thumb: Use a predicate that can easily be initialized, and is obtained by weakening the postcondition Q.
- Choose the guard B such that $J \wedge \neg B \Rightarrow Q$.

Heuristic: Split conjuncts

- If Q is of the form $Q_0 \wedge Q_1$, then it could be useful to try to isolate one conjunct as in $J \equiv Q_0$ and $B \equiv \neg Q_1$ (or vice versa).
- Clearly, J must be easy to initialize, and B must be a valid test (i.e. without specification constants).
- Sometimes Q appears to be a single conjunct while it still can be expressed as two conjuncts.
  Example: $x < y$ can be expressed as $x \leq y \wedge x \neq y$.

Heuristic: Replace expression by variable

- If Q contains an expression E, then J could be defined by replacing some (or all) occurrences of E in Q by a new variable i. This way, $J \wedge i = E \Rightarrow Q$.
- The guard must then be $B \equiv i \neq E$ and should not contain any specification constants.
- It is a good practice to augment J with some conjunct that indicates which range of values i may attain.

Heuristic: Replace constant by variable

A special case of the previous heuristic.

- If Q contains a constant n then we could define J by replacing some (or all) occurrences of n in Q by a new variable i, such that $J \wedge i = n \Rightarrow Q$.
- Clearly, the guard must be $B \equiv i \neq n$.
- Again, it is good practice to augment J with some conjunct that indicates which range of values i may attain.

Heuristic: Split a variable

A special case of the special case.

- If Q contains several occurrences of a variable k, then J could be defined by replacing some (<u>but not all</u>) occurrences of k in Q by a new variable i, such that $J \wedge i = k \Rightarrow Q$.
- Again, the guard must be $B \equiv i \neq k$.
- Again, it is good practice to augment J with some conjunct that indicates which range of values i may attain.

Heuristic: Generalization

Suppose a precondition P and a (post-regular) postcondition Q:

$$P : X = E$$
$$Q : x = X$$

Often, we can find a suitable J by generalizing E in P to some expression E_0.

- Example 1: $J : \; X = x + E$ where $\neg B \Rightarrow E = 0$.
- Example 2: $J : \; X = x \cdot E$ where $\neg B \Rightarrow E = 1$.

One could argue that this is similar to the heuristic “Replace constant by a variable” applied to the precondition:

- Example 1: $P : \; X = 0 + E$
- Example 2: $P : \; X = 1 \cdot E$

Outline

Examples: Exponentiation

Recall the following specification:

const $x : \mathbb{R}$;
var $y : \mathbb{R}$, $n : \mathbb{Z}$;
$\{P :\ n \geq 0 \land x^n = Y\}$
S
$\{Q :\ y = Y\}$

We found the invariant (and guard) by generalization:

$$J : n \geq 0 \land y \cdot x^n = Y$$
$$B : n \neq 0$$

Examples: Powers of 2

Recall the following specification:

const $x : \mathbb{Z}$;
var $i,\ y : \mathbb{Z}$;
$\quad\{P :\ x > 0\}$
T
$\quad\{Q :\ x < y \leq 2 \cdot x \wedge y = 2^i\}$

We found the invariant (and guard) by conjunct-splitting Q:

$$J : y \leq 2 \cdot x \wedge y = 2^i$$
$$B : x \geq y$$

The End

- We have covered until Section 7.3 of the reader
- Next time:
  Deriving recurrence relations for exercises in Chapter 7