COMP4011/8011
Advanced Topics in
Formal Methods and Programming Languages

– Software Verification with Isabelle/HOL –

Peter Höfner

October 6, 2024

Section 19

Invariants

Practice with Invariants

Recall:

- invariants are needed to automate the application of hoare rules
- they are used by the weakest precondition calculus to deal with loops

Recall:

- an invariant needs to be “enough” (to prove the postcondition)
- an invariant needs to be an invariant
  - “true before the loop”
  - “if true at the start of an iteration, still true after one iteration”

Weakest precondition - recall

$(P \Longrightarrow pre\ (i_0;\ i_1;\ i_2;)\ Q) \Longrightarrow \ \{\ P\ \}\ \ i_0;\ i_1;\ i_2;\ \ \{\ Q\ \}$

$\{\ P\ \}$

$pre\ i_0\ (pre\ i_1\ (pre\ i_2\ Q)) = pre\ i_1;\ i_2;\ i_3;\ Q$

$i_0;$

$pre\ i_1\ (pre\ i_2\ Q)$

$i_1;$

$pre\ i_2\ Q$

$i_2;$

$\{\ Q\ \}$

Invariant – Recall

$\{ P \}$

$P \implies I$ ("true before the loop")

?? $pre\ (WHILE\ b\ INV\ I\ DO\ c\ OD) = I$

WHILE b $INV\ I$

$I \wedge b \implies pre\ c\ I$

("if true at the start of an iteration,")

DO

("still true after one iteration")

c

OD

$I \wedge \neg b \implies Q$ ("enough")

$\{ Q \}$

Example 1

$\{ \ a \geq 0 \ \wedge \ b \geq 0 \ \}$
$A := 0;$
$B := 0;$
INV { B = b * A}
WHILE$A \neq a$
DO
    $B := B + b;$
    $A := A + 1$
OD
$\{ \ B = b * a \ \}$

$A =$	0	1	2	3	4	...
$B =$	0	b	b+b	b+b+b	b+b+b+b	...

Example 1

$\{ \; a \geq 0 \;\wedge\; b \geq 0 \; \}$
$A := 0;$
$B := 0;$ $\qquad 0 = b * 0 \quad \checkmark$
INV { B = b * A}
WHILE $A \neq a$ $\qquad B = b * A \wedge A \neq a \;\longrightarrow\; B + b = b * (A + 1)$
DO $\qquad\qquad = \; b * A \; + \; b$
$\quad B := B + b;$ $\qquad\qquad = \; B + b \quad \checkmark$
$\quad A := A + 1$
OD $\qquad B = b * A \wedge A = a \;\longrightarrow\; B = b * a \quad \checkmark$
$\{ \; B = b * a \; \}$

Example 2

$\{\ a \geq 0\ \wedge\ b \geq 0\ \}$
$A := 0;$
$B := 0;$ $\qquad 0 = b * 0 \quad \checkmark$
INV { B = b * A}
WHILE$A < a$ $\qquad B = b * A \wedge A < a \longrightarrow B + b = b * (A + 1)$
DO $\qquad = b * A + b$
$\quad B := B + b;$ $\qquad = B + b \quad \checkmark$
$\quad A := A + 1$
OD $\qquad B = b * A \wedge A \geq a \longrightarrow B = b * a \quad ???$
$\{\ B = b * a\ \}$

Example 2

$\{\ a \geq 0 \ \wedge \ b \geq 0 \ \}$
$A := 0;$
$B := 0;$ $\qquad 0 = b * 0 \wedge 0 \leq a \quad \checkmark$
INV { B = b * A ∧ A≤ a}
WHILE $A < a$ $\qquad B = b * A \wedge A < a \longrightarrow B + b = b * (A + 1)$
DO $\qquad \wedge A \leq a \qquad \wedge A + 1 \leq a \quad \checkmark$
$\quad B := B + b;$
$\quad A := A + 1$
OD $\qquad B = b * A \wedge A \geq a \longrightarrow B = b * a$
$\{\ B = b * a\ \}$ $\qquad \wedge A \leq a \quad \checkmark$

Example 3

$\{ a \geq 0 \ \wedge \ b > 0 \}$
$A := a;$
$B := 1;$

$A =$	a	a-1	a-2	a-3	...
$B =$	1	b	b*b	b*b*b	...
				$= b^3 = b^{a-A}$	

$1 = b^{a-a}$

INV $\{ B = b^{a-A} \}$
WHILE $A \neq 0$
DO
$B := B * b;$
$A := A - 1$
OD
$\{ B = b^a \}$

$B = b^{a-A} \wedge A \neq 0 \longrightarrow B * b = b^{a-(A-1)}$

$B = b^{a-A} \wedge A = 0 \longrightarrow B = b^a$

Australian National University

Example 3

$\{\ a \geq 0\ \wedge\ b > 0\ \}$
$A := a;$
$B := 1;$

$A =$	a	a-1	a-2	a-3	...
$B =$	1	b	b*b	b*b*b	...
				$= b^3 = b^{a-A}$	

$1 = b^{a-a}$

INV $\{\ B = b^{a-A}$ $\wedge$ A$\leq$ a$\}$
WHILE$A \neq 0$
DO
$B := B * b;$
$A := A - 1$
OD
$\{\ B = b^a\ \}$

$B = b^{a-A} \wedge A \neq 0 \longrightarrow\ B * b = b^{a-(A-1)}$

$B = b^{a-A} \wedge A = 0\ \longrightarrow\ B = b^a$

Example 4

$\{\ True\ \}$

$X := x;$ $\quad X = \quad [x_0; x_1; x_2 ...] \quad [x_1; x_2 ...] \quad [x_2 ...] \quad ...$

$Y := [];$ $\quad Y = \quad [] \quad x_0 \# [] \quad x_1 \# x_0 \# [] \quad ...$

$(rev\ x)@[] = rev\ x$

INV $\{\ (rev\ X)@Y = rev\ x \}$

WHILE $X \neq []$

$(rev\ X)@Y = rev\ x\ \wedge X \neq [] \longrightarrow$

$(rev\ (tl\ X))@((hd\ X)\#Y) = rev\ x$

DO

$= (rev\ X)@Y$

$\quad Y := (hd\ X\#Y);$

$= (rev\ ((hd\ X)\#(tl\ X)))@Y$

$\quad X := tl\ X$

OD

$(rev\ X)@Y = rev\ x\ \wedge X = []\ \longrightarrow Y = rev\ x$

$\{\ Y = rev\ x\ \}$

Example 5

Try with $b = 10 = 2^1 + 2^3$ or $b = 12 = 2^2 + 2^3$ (and e.g. a=3)

$\{ a \geq 0 \land b \geq 0 \}$
$A := a;\ B := b;\ C := 1;$ $\quad a^b = 1 * a^b$
INV $\{ a^b = C * A^B \}$
WHILE $B \neq 0$ $\quad a^b = C * A^B \land B \neq 0 \longrightarrow a^b = (C * A) * A^{B-1}$
DO
INV $\{ a^b = C * A^B \}$
WHILE (B mod 2 = 0)
$a^b = C * A^B \land B\ mod\ 2 = 0 \longrightarrow a^b = C * (A * A)^{B\ div\ 2}$
DO
$A := A * A;$
$B := B\ div\ 2;$
OD
$C := C * A;$
$B := B - 1$
OD $\quad a^b = C * A^B \land B = 0 \longrightarrow C = a^b$
$\{ C = a^b \}$

Example 6

$LEQ\ A\ n = \forall k.\ k < n \longrightarrow\ A!k \leq piv$
$GEQ\ A\ n = \forall k.\ n < k < length\ A \longrightarrow A!k \geq piv$
$EQ\ A\ n\ m = \forall k.\ n \leq k \leq m \longrightarrow A!k = piv$

$\{\ 0 < length\ A\ \}$
$l := 0; u := length\ A - 1; A := a$
INV $\{\ LEQ\ A\ l \wedge GEQ\ A\ u \wedge u < length\ A \wedge l \leq length\ A \wedge A$ permutes $a\}$
WHILE$l \leq u$
DO
INV $\{\ LEQ\ A\ l \wedge GEQ\ A\ u \wedge u < length\ A \wedge l \leq length\ A \wedge A$ permutes $a\}$
WHILE $l < length\ A \wedge A!l \leq piv$ DO $l := l + 1$ OD;

INV $\{\ LEQ\ A\ l \wedge GEQ\ A\ u \wedge u < length\ A \wedge l \leq length\ A \wedge A$ permutes $a\}$
WHILE $0 < u \wedge piv \leq A!u$ DO $u := u - 1$ OD;

IF $l \leq u$ THEN $A := A[l := A!u, u := A!l]$ ELSE SKIP FI
OD
$\{\ LEQ\ A\ u \wedge EQ\ A\ u\ l \wedge GEQ\ A\ l \wedge A$ permutes $a\ \}$

Example 7

Reminder:

datatype ref = Ref int | Null
Pointer access: p→field
Pointer update: p→field :== v

Definition:

"*List nxt p Ps*" is a linked list, starting at pointer *p* following the next pointer through the function *nxt*, and where *Ps* contains the list of the pointers of the linked list.

$\{\ \mathit{List\ nxt\ p\ Ps} \wedge X \in \mathit{Ps}\ \}$

$\mathrm{INV}\ \{\ \exists \mathit{Qs}.\ \mathit{List\ nxt\ p\ Qs} \wedge X \in \mathit{Qs}\}$

$\mathrm{WHILE}\ p \neq \mathit{Null} \wedge p \neq \mathit{Ref}\ \ X$

DO

$\quad p := p \rightarrow \mathit{nxt};$

OD

$\exists \mathit{Qs}.\ \mathit{List\ nxt\ p\ Qs} \wedge X \in \mathit{Qs}$

$\exists \mathit{Qs}.\ \mathit{List\ nxt\ p\ Qs} \wedge X \in \mathit{Qs}$
$\wedge p \neq \mathit{Null} \wedge p \neq \mathit{Ref}\ \ X \longrightarrow$
$\exists \mathit{Qs}.\ \mathit{List\ nxt}\ (p \rightarrow \mathit{nxt})\ \mathit{Qs} \wedge X \in \mathit{Qs}$

$\exists \mathit{Qs}.\ \mathit{List\ nxt\ p\ Qs} \wedge X \in \mathit{Qs}$
$\wedge (p = \mathit{Null} \vee p = \mathit{Ref}\ \ X) \longrightarrow\ \ p = \mathit{Ref}\ \ X$

Example 8

What is Isabelle function doing?

fun f :: $'a\ list \Rightarrow' a\ list \Rightarrow' a\ list$ **where**
$f\ []\ ys = ys\ |$
$f\ xs\ []\ = xs\ |$
$f\ (x\#xs)\ (y\#ys) = x\#y\#\ f\ xs\ ys$

Example 8

What is Isabelle function doing?

$$\textbf{fun}\ \mathit{splice}\ ::\ 'a\ list \Rightarrow 'a\ list \Rightarrow 'a\ list\ \textbf{where}$$
$$\mathit{splice}\ []\ ys = ys\ |$$
$$\mathit{splice}\ xs\ []\ = xs\ |$$
$$\mathit{splice}\ (x\#xs)\ (y\#ys) = x\#y\#\ f\ xs\ ys$$

Let's write it with linked lists!

Example 8

$List\ nxt\ p\ Ps = Path\ nxt\ p\ Ps\ Null$

$Path\ nxt\ p\ Ps\ Null$ is a linked list from p to q following function nxt and containing list of pointers Ps

$\{\ List\ nxt\ p\ Ps \wedge List\ nxt\ q\ Qs \wedge (set\ Ps \cap set\ Qs) = \{\} \wedge size\ Qs \leq size\ Ps\ \}$

$pp := p;$

INV $\{\ \exists PPs\ QQs\ PPPs.\ \ size\ QQs \leq size\ PPs \wedge$

$List\ nxt\ pp\ PPs\ \wedge List\ nxt\ q\ QQs\ \wedge Path\ nxt\ p\ PPPs\ pp$

$\wedge\ PPPs@splice\ PPs\ QQs\ =\ splice\ Ps\ Qs\ \wedge$

$set\ PPs \cap set\ QQs = \{\} \wedge distinct\ PPPs \wedge set\ PPPs \cap (set\ PPs \cup set\ QQs) = \{\}$

$\}$

WHILE$q \neq Null$

DO

$qq := q \rightarrow nxt; q \rightarrow nxt := pp \rightarrow nxt; pp \rightarrow nxt = q; pp := q \rightarrow nxt; q := qq;$

OD

$\{\ List\ nxt\ p\ (splice\ Ps\ Qs)\ \}$

Demo