Formal Verification – Inference Rules

① $$\frac{}{\llparenthesis Q[E/x] \rrparenthesis \quad x = E \quad \llparenthesis Q \rrparenthesis}$$

assignment. We typically apply this rule from the end of a program forward.

② $$\frac{\underbrace{(\overset{\text{stronger}}{P} \to \overset{\text{weaker}}{P'})}_{\text{requires a separate proof}} \qquad \llparenthesis P' \rrparenthesis\ C\ \llparenthesis Q \rrparenthesis}{\llparenthesis P \rrparenthesis\ C\ \llparenthesis Q \rrparenthesis}$$

"precondition strengthening"

implied. requires a separate natural deduction proof.

③ $$\frac{\llparenthesis P \rrparenthesis\ C\ \llparenthesis Q' \rrparenthesis \qquad \underbrace{(\overset{\text{stronger}}{Q'} \to \overset{\text{weaker}}{Q})}_{\text{requires a separate proof}}}{\llparenthesis P \rrparenthesis\ C\ \llparenthesis Q \rrparenthesis}$$

"post condition weakening"

implied. requires a separate natural deduction proof.

④ $$\frac{\llparenthesis P \rrparenthesis\ C_1\ \llparenthesis Q \rrparenthesis \qquad \llparenthesis Q \rrparenthesis\ C_2\ \llparenthesis R \rrparenthesis}{\llparenthesis P \rrparenthesis\ C_1 ; C_2\ \llparenthesis R \rrparenthesis}$$

composition. We only use this rule implicitly, and never cite it.

⑤ $$\frac{\llparenthesis P \land B \rrparenthesis\ C_1\ \llparenthesis Q \rrparenthesis \qquad \llparenthesis (P \land (\neg B)) \rrparenthesis\ C_2\ \llparenthesis Q \rrparenthesis}{\llparenthesis P \rrparenthesis\ \boxed{\text{if } (B)\ C_1 \text{ else } C_2}\ \llparenthesis Q \rrparenthesis}$$

if–then–else

note this difference!

⑥ $$\frac{\llparenthesis P \land B \rrparenthesis\ C\ \llparenthesis Q \rrparenthesis \qquad ((P \land (\neg B)) \to Q)}{\llparenthesis P \rrparenthesis\ \boxed{\text{if } (B)\ C}\ \llparenthesis Q \rrparenthesis}$$

if–then

requires a separate proof.

⑦ $$\frac{\llparenthesis (\overset{\text{invariant}}{I} \land B) \rrparenthesis\ C\ \llparenthesis I \rrparenthesis}{\llparenthesis I \rrparenthesis\ \boxed{\text{while } (B)\ C}\ \llparenthesis (I \land (\neg B)) \rrparenthesis}$$

partial-while

key: ① identify the invariant.
② show that the precondition implies the invariant
③ show that the invariant implies the postcondition

Solutions

Assignments

Complete the following annotations

$Q[E/x] = (2=2)$

① $\{(2=2)\}$
$x = 2;$ E is 2.
$\{(x=2)\}$ assignment
Q is $(x=2)$

② $\{(y=2)\}$
$x = 2;$
$\{(x=y)\}$ assignment

③ $\{(0=2)\}$
$x = 2;$
$\{(x=0)\}$ assignment

$Q[E/x] = Q[(x+1)/x] = (x+1 = n+1)$

④ $\{(x+1 = n+1)\}$
$x = x+1;$ $E = x+1$
$\{x = n+1\}$ assignment
Q is $x = n+1$.

$Q[E/x] = [y/x] = (2 \cdot y = y + y)$

⑤ $\{(2 \cdot y = y + y)\}$
$x = y;$ $E = y$.
$\{(2 \cdot x = x + y)\}$ assignment
Q is $(2 \cdot x = x + y)$.

The "assignment" inference rule

$\{Q[E/x]\}$ then Q must be true when replacing every x by E
$x = E;$
$\{Q\}$ if Q is true after assigning x to E,

Notes

① E may contain x in it. Treat E as a whole expression and do not worry about what's inside.

② When writing down $Q[E/x]$, do NOT change the order of things in the formula and do NOT simplify it.

solutions

Assignments

Prove that the following program satisfies the given triple under partial correctness

Steps:

1. Push up assignments
2. Prove any implied's.

① $(\!|\,((x = x_0) \wedge (y = y_0))\,|\!)$

$(\!|\,((y = y_0) \wedge (x = x_0))\,|\!)$ implied

$t = x;$

$(\!|\,((y = y_0) \wedge (t = x_0))\,|\!)$ assignment

$x = y;$

$(\!|\,((x = y_0) \wedge (t = x_0))\,|\!)$ assignment

$y = t;$

$(\!|\,((x = y_0) \wedge (y = x_0))\,|\!)$ assignment

Proof of implied: $\vdash (((x = x_0) \wedge (y = y_0)) \to ((y = y_0) \wedge (x = x_0)))$

1.	$(x = x_0) \wedge (y = y_0)$	assumption
2.	$x = x_0$	$\wedge e$: 1
3	$y = y_0$	$\wedge e$: 1
4.	$(y = y_0) \wedge (x = x_0)$	$\wedge i$: 2, 3

5. $(((x = x_0) \wedge (y = y_0)) \to ((y = y_0) \wedge (x = x_0)))$ $\to i$: 1–4

② $(\!|\,\text{true}\,|\!)$

$(\!|\,(x + y = x + y)\,|\!)$ implied

$z = x;$

$(\!|\,(z + y = x + y)\,|\!)$ assignment

$z = z + y;$

$(\!|\,(z = x + y)\,|\!)$ assignment

$u = z;$

$(\!|\,(u = x + y)\,|\!)$ assignment

Proof of implied: $\vdash (\text{true} \to (x + y = x + y))$

1.	true	assumption
2	$(x + y = x + y)$	EQ1 + $\forall e$.

3. $(\text{true} \to (x + y = x + y))$ $\to i$: 1–2

Conditional Statement (if-then) & (if-then-else)

① "if-then"

$\{ P \}$

if (B) {

  $\{ (P \wedge B) \}$ if-then ①

  $\{ P' \}$ implied (a) ②

  C

  $\{ Q \}$ [justify based on C] ②

}

$\{ Q \}$ if-then

  implied (b) $((P \wedge (\neg B)) \rightarrow Q)$ } ①

③ { proof of implied (a)
  proof of implied (b)

Steps:
① Annotate the if-then's
② Push up assignments
③ Prove any implied's

② "if-then-else"

$\{ P \}$

if (B) {

  $\{ (P \wedge B) \}$ if-then-else ①

  $\{ P_1 \}$ implied (a) ②

  C_1

  $\{ Q \}$ [justify based on C_1] ②

} else {

  $\{ (P \wedge (\neg B)) \}$ if-then-else ①

  $\{ P_2 \}$ implied (b) ②

  C_2

  $\{ Q \}$ [justify based on C_2] ②

}

$\{ Q \}$ if-then-else ①

③ Proofs of implied (a) and (b)

solutions

Common Questions: about if-then:

① Why did you annotate the last line with "implied" and the implication $((P \wedge (\neg B)) \rightarrow Q)$?

For an if-then statement, there are 2 cases to consider. If B is true, then we go inside the if-block and execute C. If B is false, we skip the if-block. The last "implied" annotation takes care of the second case. Even when we skip the if-block, we still need to show that Q is satisfied. This corresponds to proving the implication $((P \wedge (\neg B)) \rightarrow Q)$.

② How did you get P' and why do you know that $(P \wedge B)$ implies P'?

I derived P' by looking at C and Q and figuring out what precondition needs to be satisfied if Q is true after executing C. For example, if C consists of assignments only, then P' is the result of pushing up Q through C.

I don't know $(P \wedge B)$ implies P'. By writing down P' and "implied" as the justification, I am saying that: "if this program satisfies my specification, then I need to prove that $(P \wedge B)$ implies P'."

Conditional Statements (If-Then)

Prove that the following program satisfies the given triple under partial correctness

About line 3: you can immediately simplify it and write down $(max < x)$ instead.

1 $(\!|\ \mathrm{true}\ |\!)$

2 if $(max < x)$ {

3 $(\!|(\mathrm{true} \wedge (max < x))|\!)$ if-then

4 $(\!|(x \geq x)|\!)$ implied (a)

5 $max = x;$

6 $(\!|(max \geq x)|\!)$ assignment

7 }

8 $(\!|(max \geq x)|\!)$ if-then

implied (b) $((\mathrm{true} \wedge (\neg(max < x))) \rightarrow (max \geq x))$

① Proof of implied (a): $\vdash ((\mathrm{true} \wedge (max < x)) \rightarrow (x \geq x))$

(This is an informal proof. See a formal proof on the next page.)

$(x \geq x)$ is a tautology. Thus the implication holds.

② Proof of implied (b): $\vdash ((\mathrm{true} \wedge (\neg(max < x))) \rightarrow (max \geq x))$

1.	$(\mathrm{true} \wedge (\neg(max < x)))$	assumption
2	$(\neg(max < x))$	$\wedge e$: 1
3	$(max \geq x)$	def. of $\geq$
4	$((\mathrm{true} \wedge (\neg(max < x))) \rightarrow (max \geq x))$	$\rightarrow i$: 1–3

solutions

Proof of implied (a): $\vdash ((\text{true} \wedge (max < x)) \to (x \leq x))$

1.	$(\text{true} \wedge (max < x))$	assumption
2.	$(x + 0 = x)$	PA3 + $\forall e$
3.	$\exists z\, (x + z = x)$	$\exists i$: 2.
4.	$(x \leq x)$	def. of $\leq$
5.	$((\text{true} \wedge (max < x)) \to (x \leq x))$	$\to i$: 1–4.

Conditional Statements (If-Then-Else)

Prove that the following program satisfies the given triple under partial correctness.

$\llparenthesis\ true\ \rrparenthesis$

$if\ (x > y)\ \{$

$\quad \llparenthesis (true \land (x > y)) \rrparenthesis$ — if-then-else

$\quad \llparenthesis (((x > y) \land (x = x)) \lor ((x \leq y) \land (x = y))) \rrparenthesis$ — implied (a)

$\quad max = x;$

$\quad \llparenthesis (((x > y) \land (max = x)) \lor ((x \leq y) \land (max = y))) \rrparenthesis$ — assignment.

$\}\ else\ \{$

$\quad \llparenthesis (true \land (\neg (x > y))) \rrparenthesis$ — if-then-else

$\quad \llparenthesis (((x > y) \land (y = x)) \lor ((x \leq y) \lor (y = y))) \rrparenthesis$ — implied (b).

$\quad max = y;$

$\quad \llparenthesis (((x > y) \land (max = x)) \lor ((x \leq y) \land (max = y))) \rrparenthesis$ — assignment.

$\}$

$\llparenthesis ((x > y) \land (max = x)) \lor ((x \leq y) \land (max = y)) \rrparenthesis$ — if-then-else

Proof of implied (a):

1. $(x > y)$ — premise.
2. $(x = x)$ — EQ I + $\forall e$
3. $((x > y) \land (x = x))$ — $\land i$: 1, 2
4. $(((x > y) \land (x = x)) \lor ((x \leq y) \land (x = y)))$ — $\lor i$: 3