Formal Verification – Inference Rules

① $$\frac{}{(\!|\, Q[E/x] \,|\!)\ \ x = E\ \ (\!|\, Q \,|\!)}$$

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

② $$\frac{(\overset{\text{stronger}}{P} \rightarrow \overset{\text{weaker}}{P'})\qquad (\!|\, P' \,|\!)\ C\ (\!|\, Q \,|\!)}{(\!|\, P \,|\!)\ C\ (\!|\, Q \,|\!)}$$

requires a separate proof

"precondition strengthening"

implied requires a separate natural deduction proof.

③ $$\frac{(\!|\, P \,|\!)\ C\ (\!|\, Q' \,|\!)\qquad (\overset{\text{stronger}}{Q'} \rightarrow \overset{\text{weaker}}{Q})}{(\!|\, P \,|\!)\ C\ (\!|\, Q \,|\!)}$$

requires a separate proof.

"post condition weakening"

implied requires a separate natural deduction proof.

④ $$\frac{(\!|\, P \,|\!)\ C_1\ (\!|\, Q \,|\!)\qquad (\!|\, Q \,|\!)\ C_2\ (\!|\, R \,|\!)}{(\!|\, P \,|\!)\ C_1 ; C_2\ (\!|\, R \,|\!)}$$

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

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

if–then–else

note this difference!

⑥ $$\frac{(\!|\, P \land B \,|\!)\ C\ (\!|\, Q \,|\!)\qquad ((P \land (\neg B)) \rightarrow Q)}{(\!|\, P \,|\!)\ \boxed{\text{if } (B)\ C}\ (\!|\, Q \,|\!)}$$

if–then

requires a separate proof.

⑦ $$\frac{(\!|\, (\overset{\text{invariant,}}{I} \land B) \,|\!)\ C\ (\!|\, I \,|\!)}{(\!|\, I \,|\!)\ \boxed{\text{while } (B)\ C}\ (\!|\, (I \land (\neg B)) \,|\!)}$$

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 $\{(2=2)\}$ ② $\{(2=y)\}$ ③ $\{(2=0)\}$

$x = 2;$ E is 2 $x = 2;$ $x = 2;$

$\{(x=2)\}$ assignment $\{(x=y)\}$ assignment $\{(x=0)\}$ assignment

Q is $(x=2)$

$Q[E/x] = Q[(x+1)/x] = (x+1 = n+1)$ $Q[E/x] = [y/x] = (2\cdot y = y+y)$

④ $\{(x+1 = n+1)\}$ ⑤ $\{(2\cdot y = y+y)\}$

$x = x+1;$ $E = x+1$ $x = y;$ $E = y.$

$\{x = n+1\}$ assignment $\{(2\cdot x = x+y)\}$ assignment

Q is $x = n+1.$ 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.

Assignments

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

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

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

$t = x;$

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

$x = y;$

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

$y = t;$

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

Steps:

① Push up assignments

② Prove any implied's.

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

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

5. $(((x = x_0) \land (y = y_0)) \to ((y = y_0) \land (x = x_0)))$ $\to i$: 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

Solutions

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

① "if-then"

```
⦃ P ⦄
if ( B ) {
    ⦃ (P ∧ B) ⦄          if-then ①
    ⦃ P' ⦄               implied (a) ②
    C
    ⦃ Q ⦄                [justify based on C] ②
}
⦃ Q ⦄                    if-then
                         implied (b)  ((P ∧ (¬B)) → Q)  } ①
```

Steps:

① Annotate the if-then's

② Push up assignments

③ Prove any implied's

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

② "if-then-else"

```
⦃ P ⦄
if ( B ) {
    ⦃ (P ∧ B) ⦄          if-then-else ①
    ⦃ P₁ ⦄               implied (a) ②
    C₁
    ⦃ Q ⦄                [justify based on C₁] ②
} else {
    ⦃ (P ∧ (¬B)) ⦄       if-then-else ①
    ⦃ P₂ ⦄               implied (b) ②
    C₂
    ⦃ Q ⦄                [justify based on C₂] 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 $\llparenthesis \text{true} \rrparenthesis$

2 if $(max < x)$ {

3 $\llparenthesis (\text{true} \land (max < x)) \rrparenthesis$ if-then

4 $\llparenthesis (x \geq x) \rrparenthesis$ implied (a)

5 max = x;

6 $\llparenthesis (max \geq x) \rrparenthesis$ assignment

7 }

8 $\llparenthesis (max \geq x) \rrparenthesis$ if-then

implied (b) $((\text{true} \land (\neg(max < x))) \rightarrow (max \geq x))$

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

(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): $\vdash ((\text{true} \land (\neg(max < x))) \rightarrow (max \geq x))$

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

Proof of implied (a): $\vdash ((true \wedge (max < x)) \rightarrow (x \leq x))$

1.	$(true \wedge (max < x))$	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.	$((true \wedge (max < x)) \rightarrow (x \leq x))$	$\rightarrow i$: 1–4.

Conditional Statements (If-Then-Else)

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

$\llparenthesis \text{true} \rrparenthesis$

```
if (x > y) {
```

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

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

```
    max = x;
```

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

```
} else {
```

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

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

```
    max = y;
```

$\llparenthesis (((x > y) \land (max = x)) \lor ((x \leq y) \land (max = y))) \rrparenthesis$ — 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. $(x = x)$ — EQI + $\forall e$
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