Hoare logic

(for the correctness of computer program; proposed by Tony Hoare in 1969)

- For a given program, how can we be sure that the program always produces the correct answer?

- After all the bugs have been removed so that the syntax is correct, we can test the program with sample input.

- But even if the program gives the correct answer for all sample input, it may not always produce the correct answer (unless all possible input has been tested).

We need a proof to show that the program always gives the correct output.

-A proof that a program is correct consists of two parts-

1. The first part shows that the correct answer is obtained if the program terminates. This part of the proof establishes the partial correctness of the program.

2. The second part of the proof shows that the program always terminates.

To prove the correctness of proof, two propositions are used.

$P \{S\} Q$

The first is the initial assertion, which gives the properties that the input values must have.

The second is the final assertion, which gives the properties that the output of the program should have, if the program did what was intended.

The appropriate initial and final assertions

must be provided when a program is checked.

DEFINITION 1 A program, or program segment, S is said to be *partially correct with respect to* the initial assertion p and the final assertion q if whenever p is true for the input values of S and S terminates, then q is true for the output values of S. The notation $p\{S\}q$ indicates that the program, or program segment, S is partially correct with respect to the initial assertion p and the final assertion q.

p is true, S terminates ⇒ q is true for S.
(for S)

P{S}Q → notation for partially correctness.

Hoare triple

Show that the program segment

$y := 2$
$z := x + y$

is correct with respect to the initial assertion p: $x = 1$ and the final assertion q: $z = 3$.

Suppose p is true for S; i.e. $x=1$ in S, when S runs.

Then $y = 2$ and $z = 1 + 2 = 3$

Also q: $z = 3$, which is true for S. Thus P{S}Q is true.

RULES OF INFERENCE

A useful rule of inference proves that a program is correct by splitting the program into a sequence of subprograms and then showing that each subprogram is correct.

Composition rule

Suppose that the program S is split into subprograms S1 and S2.
Write S = S1; S2 to indicate that S is made up of S1 followed by S2.

Suppose that the correctness of S1 with respect to the initial assertion p and final assertion q,

and the correctness of S2 with respect to the initial assertion q and the final assertion r,

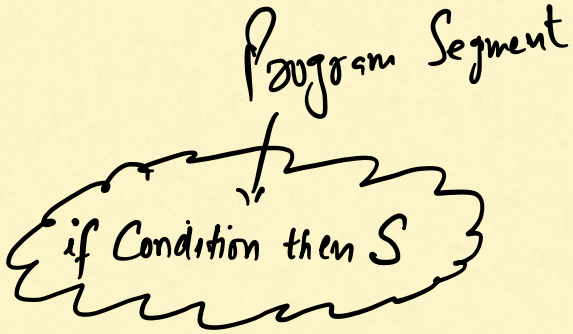
have been established.

It follows that if p is true and S1 is executed and terminates, then q is true; and if q is true, and S2 executes and terminates, then r is true. Thus, if p is true and S = S1; S2 is executed and terminates, then r is true.

This rule of inference, called the composition rule, can be stated as:

$$\frac{p\ \{S_1\}\ q \qquad q\ \{S_2\}\ r}{\therefore\ p\ \{S_1; S_2\}\ r.}$$

Conditional Statements



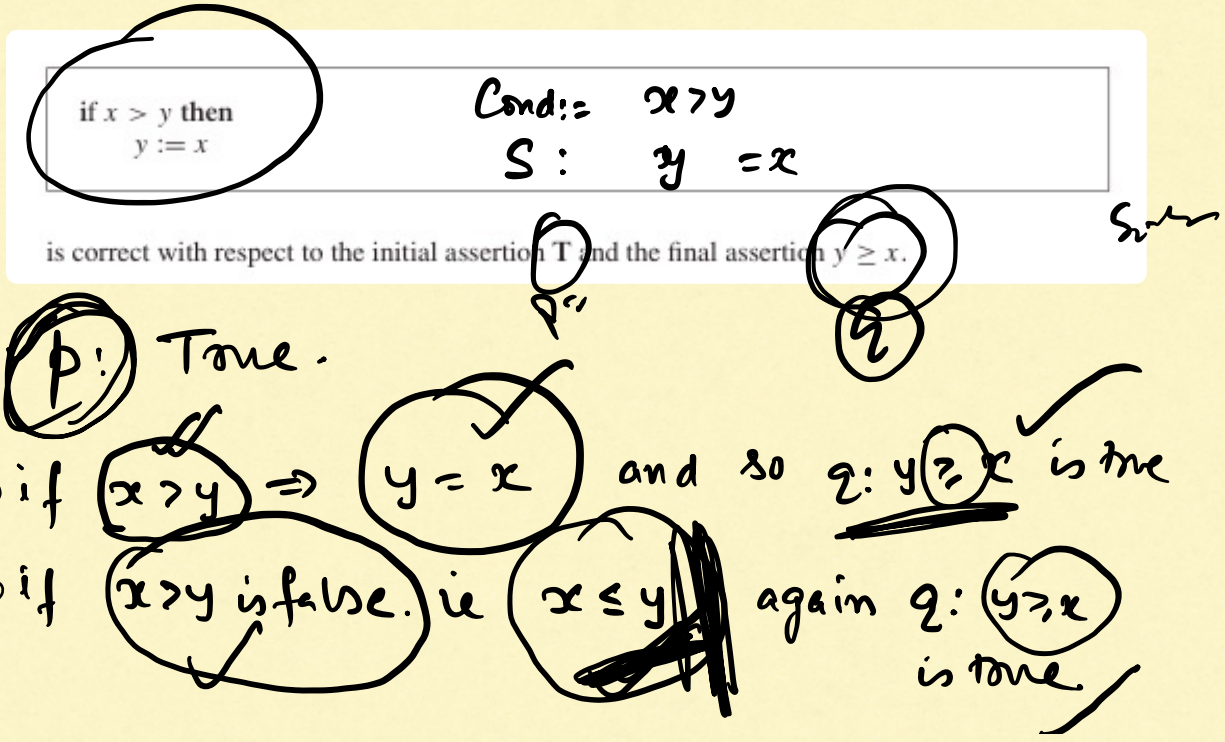
$$\frac{(p \wedge \text{Condition})\ \{S\}\ q \quad \text{(part 1)} \qquad (p \wedge \sim \text{Condition}) \rightarrow q \quad \text{(part 2)}}{\therefore\ p\ \{\text{if Condition then } S\}\ q \quad \text{(final)}}$$

S is executed if condition is true,
and it is not executed when condition is false.

To verify that this segment is correct with respect to the initial assertion p and final assertion q , two things must be done.

First, it must be shown that when p is true and condition is also true, then q is true after S terminates.

Second, it must be shown that when p is true and condition is false, then q is true (because in this case S does not execute).



if Condition then S_1 else S_2.

Exercise

$(p \wedge \text{Condition})\ \{S_1\}\ q$

$(p \wedge \neg \text{Condition})\ \{S_2\}\ q$

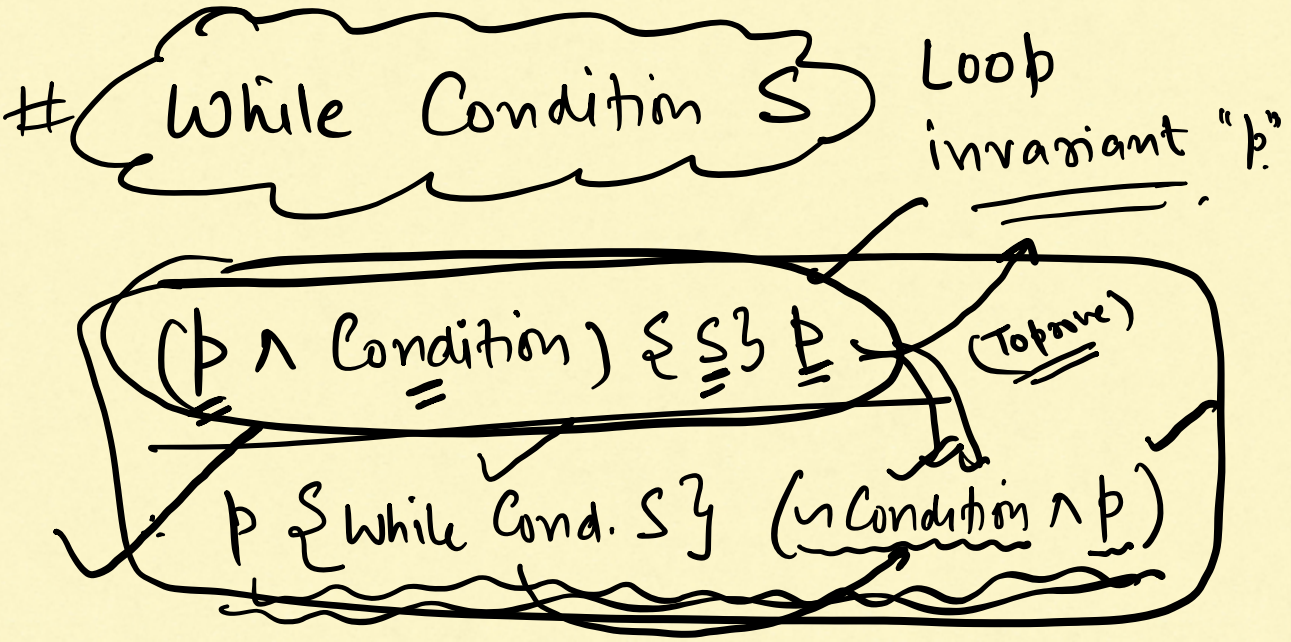
$\therefore\ p\ \{\text{if Condition then } S_1 \text{ else } S_2\}\ q$ ✓

Verify that the program segment

```
if x < 0 then        ← Cond.
    abs := −x        ← S1
else
    abs := x         ← S2
```

is correct with respect to the initial assertion T and the final assertion $abs = |x|$.

Pf: Exercise.



S executed repeatedly until Cond. becomes false.

p is true each time S executes.

p is loop invariant assertion

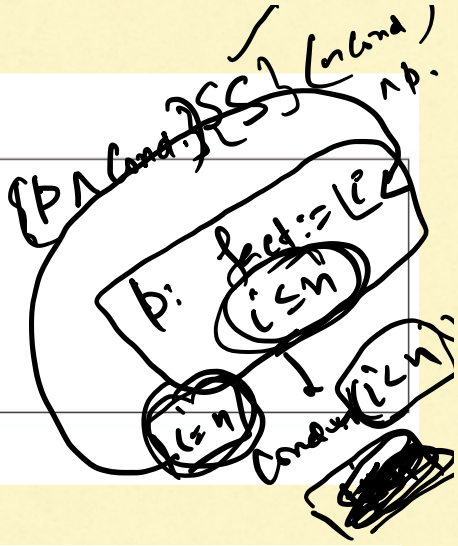
if (p ∧ Condition) {S} (p) is true.

And after termination

p ∧ ¬Condition is true

A loop invariant is needed to verify that the program segment

```
i := 1
factorial := 1
while i < n
    i := i + 1
    factorial := factorial · i
```

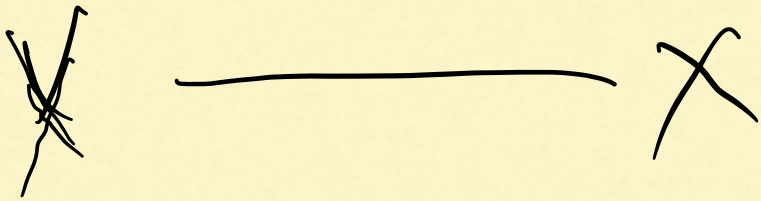


terminates with $factorial = n!$ when n is a positive integer.

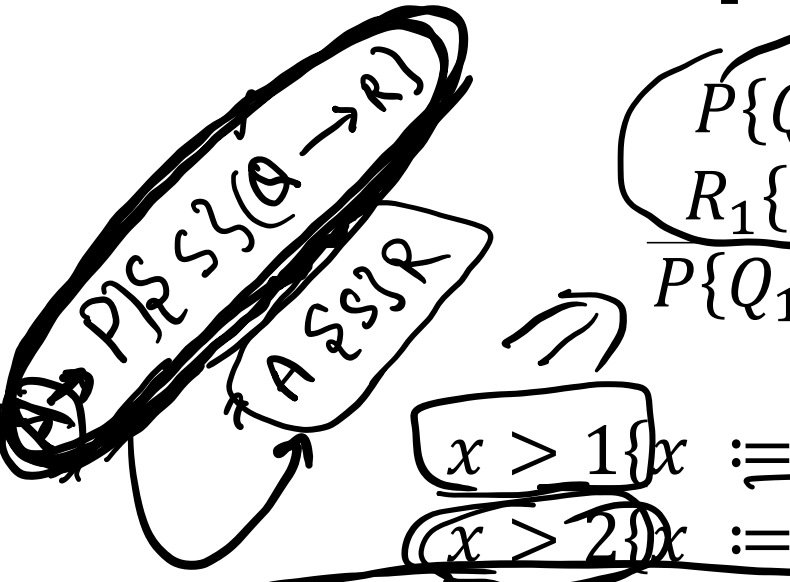
Let p be the assertion "$factorial = i!$ and $i \leq n$." We first prove that p is a loop invariant. Suppose that, at the beginning of one execution of the **while** loop, p is true and the condition of the **while** loop holds; in other words, assume that $factorial = i!$ and that $i < n$. The new values i_{new} and $factorial_{new}$ of i and $factorial$ are $i_{new} = i + 1$ and $factorial_{new} = factorial \cdot (i + 1) = (i + 1)! = i_{new}!$. Because $i < n$, we also have $i_{new} = i + 1 \leq n$. Thus, p is true at the end of the execution of the loop. This shows that p is a loop invariant.

Now we consider the program segment. Just before entering the loop, $i = 1 \leq n$ and $factorial = 1 = 1! = i!$ both hold, so p is true. Because p is a loop invariant, the rule of inference just introduced implied that if the **while** loop terminates, it terminates with p true and with $i < n$ false. In this case, at the end, $factorial = i!$ and $i \leq n$ are true, but $i < n$ is false; in other words, $i = n$ and $factorial = i! = n!$, as desired.

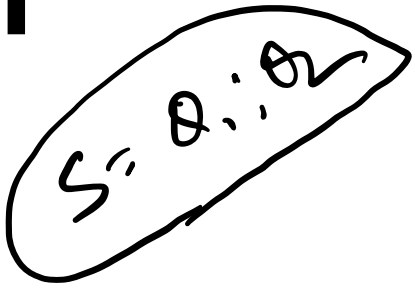
Finally, we need to check that the **while** loop actually terminates. At the beginning of the program i is assigned the value 1, so after $n - 1$ traversals of the loop, the new value of i will be n, and the loop terminates at that point.



Composition



$$\frac{P\{Q_1\}R_1 \quad R_1\{Q_2\}R}{P\{Q_1; Q_2\}R}$$



$$\frac{x > 1\{x := x + 1\}x > 2 \quad x > 2\{x := x + 1\}x > 3}{x > 1\{x := x + 1; x := x + 1\}x > 3}$$

$$\frac{x > 1\{x := x + 1\}x > 2 \quad x > 0\{x := -x\}x < 0}{x > 1\{x := x + 1; x := -x\}\ ?}$$

Consequence

$$\frac{P\{Q\}R \qquad R \to S}{P\{Q\}S} \qquad \frac{P\{Q\}R \qquad S \to P}{S\{Q\}R}$$

$$\frac{\dfrac{x > 1\{x := x + 1\}x > 2 \qquad x > 2 \to x > 0}{x > 1\{x := x + 1\}x > 0} \qquad x > 0\{x := -x\}x < 0}{x > 1\{x := x + 1; x := -x\}x < 0}$$