Hoare Logic: An Introduction

What is Hoare Logic?

- A formal system for reasoning about program correctness.
- Uses **Hoare triples** of the form

$$\{P\}C\{Q\}$$

where:

- P is the **precondition** (what must be true before execution).
- C is the **command** (the program to execute).
- Q is the **postcondition** (what is guaranteed after execution).

Hoare Triples: Meaning

$$\{x = 2\}(x := x + 1)\{x = 3\}$$

or

$$\begin{array}{c} \{x = 2\} \\ x := x + 1 \\ \{x = 3\} \end{array}$$

- Before executing $x := x + 1$, x is 2.
- After execution, x is guaranteed to be 3.

The Axioms and Rules of Hoare Logic

Assignment Rule

If P holds after replacing x with E, then P holds after assignment.

$$\{P[x := E]\}x := E\{P\}$$

Example

$$\{y + 1 = 5\}x := y + 1\{x = 5\}$$

The Rules (Continued)

Consequence Rule

If P' implies P, and Q implies Q', then the triple is valid.

$$\frac{\{P\}C\{Q\}, P' \to P, Q \to Q'}{\{P'\}C\{Q'\}}$$

Composition Rule

If $C1$ ensures Q and $C2$ ensures R, then their sequence ensures R.

$$\frac{\{P\}C1\{Q\}, \{Q\}C2\{R\}}{\{P\}C1; C2\{R\}}$$

Hoare Logic for Conditionals

If-Else Rule

The postcondition Q holds regardless of which branch executes.

$$\frac{\{P \wedge B\}C1\{Q\}, \{P \wedge \neg B\}C2\{Q\}}{\{P\}\text{if } B \text{ then } C1 \text{ else } C2\{Q\}}$$

Example:

$$\{x > 0\}$$
$$\text{if } x > 0 \text{ then } y := x \text{ else } y := 0$$
$$\{y \geq 0\}$$

Hoare Logic for Loops

While Loop Rule

$$\frac{\{P \wedge B\} C \{P\}}{\{P\} \text{while } B \text{ do } C \{P \wedge \neg B\}}$$

- **Loop invariant** P remains true before and after each iteration.
- Ensures correctness of loops by proving invariance.

Example: Proving a Simple Loop

Program:

```
{ x = n, y = 0 }
while y < n do
  y := y + 1;
  x := x - 1
{ y = n }
```

Loop Invariant:

```
{ x + y = n }
```

- Initially true ($\texttt{y = 0}, \texttt{x = n} \rightarrow \texttt{x + y = n}$).
- Maintained after each iteration.
- When loop exits ($\texttt{x = 0, y = n}$) holds.

Summary

- **Hoare Logic** allows formal reasoning about program correctness.
- Uses **Hoare triples** $\{P\}C\{Q\}$ to specify behavior.
- Key rules:
  - Assignment
  - Composition
  - Conditionals
  - Loops
- Loop invariants are crucial for proving correctness.