Hoare Logic: An Introduction

What is Hoare Logic?

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

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

or

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

 ${x = 3}$

- ▶ 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.

$\{P\}C1\{Q\},\{Q\}C2\{R\}$	ľ
$\{P\}C1; C2\{R\}$	_

Hoare Logic for Conditionals

If-Else Rule

The postcondition Q holds regardless of which branch executes.

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

Example:

Hoare Logic for Loops

While Loop Rule

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

- ▶ **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\}
```

- lnitially true $(y = 0, x = n \rightarrow x + y = n)$.
- ► Maintained after each iteration.
- ▶ When loop exits (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.