Reasoning About Programs

- What is true of a program’s state as it executes?
  - Given initial assumption or final goal

- Examples:
  - If $x > 0$ initially, then $y == 0$ when loop exits
  - Contents of array are sorted
  - Except at one program point, $x + y == z$
  - For all instances of Node $n$,
    - $n.next == \text{nil} \lor n.next.prev == n$
    - ...

Forward Reasoning Example

- Suppose we initially know (or assume) $w > 0$

```plaintext
// w > 0
x = 17;
// w > 0 \land x == 17
y = 42;
// w > 0 \land x == 17 \land y == 42
z = w + x + y;
// w > 0 \land x == 17 \land y == 42 \land z > 59
...
```

Then we know various things after, e.g., $z > 59$

Backward Reasoning Example

- Suppose we want $z < 0$ at the end

```plaintext
// w + 17 + 42 < 0
x = 17;
// w + x + 42 < 0
y = 42;
// w + x + y < 0
z = w + x + y;
// z < 0
```

Then initially we need $w < -59$
Forward vs. Backward

- Forward Reasoning
  - Determine what follows from initial assumptions
  - Useful for ensuring an invariant is maintained

- Backward Reasoning
  - Determine sufficient conditions for a certain result
    - Desired result: assumptions need for correctness
    - Undesired result: assumptions needed to trigger bug
  - Less natural but often more useful

Pre- and PostConditions

Precondition → \{ w < 59 \}
\hspace{1cm} x = 17;

Postcondition → \{ x = 17 \land w + x < -42 \}

- An assertion holds if evaluating it in the current state produces true.

Conditional Example (Fwd)

// x >= 0
z = 0;
// x >= 0 ∧ z = 0
if (x != 0) {
    // x >= 0 ∧ z = 0 ∧ x != 0 (so x > 0)
    z = x;
    // ... ∧ z > 0
} else {
    // x >= 0 ∧ z = 0 ∧ !(x != 0) (so x = 0)
    z = x + 1;
    // ... ∧ z = 1
}
// (... ∧ z > 0) ∨ (... ∧ z = 1) (so z > 0)

Hoare Triples

\{ P \} S \{ Q \}

- Hoare triple \{P\} S \{Q\} is valid iff:
  - For all states where P holds, executing S always produces a state where Q holds

  “If P is true before S, then Q must be true after”
**Hoare Triple Examples**

- Valid or invalid?
  - Assume all variables are integers without overflow

  - \{x \neq 0\} \ y = x*x; \ {y > 0}\ 
  - \{z \neq 1\} \ y = z*z; \ {y \neq z}\ 
  - \{x \geq 0\} \ y = 2*x; \ {y > x}\ 
  - \{true\} \ 
    - if (x > 7) \{ y=4; \} else \{ y=3; \}\ 
    - \{true\} \ 
      - x = y; \ z = x; \ {y=z}\ 
  - \{x=7 \land y=5\} \ tmp = x; \ x = tmp; \ y = x; \ {y=7 \land x=5}\ 

**Assignment**

\{ P \} \ x = e; \ { Q \} 

- Valid if: \ P \Rightarrow Q[x:=e] 

  - Example: \{z > 34\} \ y = z + 1; \ {y > 1}\ 
    - Valid: \{z > 34\} \Rightarrow \{z + 1 > 1\}

**Sequence**

\{ P \} S1; S2 \{ Q \}

- Valid if: there is an R such that:
  1. \{ P \} S1 \{ R \}
  2. \{ R \} S2 \{ Q \}

  - Example:
    - \{z \geq 1\}\ 
      - \ y = z + 1; \ w = y * y; \ {w > y}\ 

**Conditional**

\{ P \} if(b) S1 else S2 \{ Q \}

- Valid if: there are Q1, Q2:
  1. \{ P \land b \} S1 \{ Q1 \}
  2. \{ P \land \neg b \} S2 \{ Q2 \}
  3. Q1 \lor Q2 implies Q

  - Example:
    - \{ true \} \ 
      - if (x > 7) \ y = x; \ 
      - else \ y = 20; \ { y > 5 }\ 
    - Q1 = \{y > 7\}. \ Q2 = \{y = 20\}
      1. \{true \land x > 7\} \ y = x \ {y > 7}\ 
      2. \{true \land x \leq 7\} \ y = 20 \ \{y = 20\}
      3. \ y > 7 \lor y = 20 \Rightarrow y > 5
Conditional

\{ true \}
\textbf{if} (x > 7)
\{ true \}/ \{ x > 7 \}
y = x;
\{ y > 7 \}
\textbf{else}
\{ true \}/ \{ x \leq 7 \}
y = 20;
\{ y = 20 \}
\{ y > 5 \}

Weaker vs. Stronger Assertions

• If P1 => P2 then:
  – P1 is stronger than P2
  – P2 is weaker than P1

• Whenever P1 holds, P2 is guaranteed to hold

Strength and Hoare Logic

Suppose \{ P1 \} S \{ Q1 \}. \{ x > 0 \} S \{ y < 5 \}
Strength and Hoare Logic

P2 => P1

Suppose \{P1\} S \{Q1\}.  \{x>0\} S \{y<5\}

• \{P2\} S \{Q1\}  \{x>10\} S \{y<5\}

Strength and Hoare Logic

Q1 => Q2

Suppose \{P1\} S \{Q1\}.  \{x>0\} S \{y<5\}

• \{P2\} S \{Q1\}  \{x>10\} S \{y<5\}

• \{P1\} S \{Q2\}  \{x>0\} S \{y<10\}

Strength and Hoare Logic

Q2

Suppose \{P1\} S \{Q1\}.  \{x>0\} S \{y<5\}

• \{P2\} S \{Q1\}  \{x>10\} S \{y<5\}

• \{P1\} S \{Q2\}  \{x>0\} S \{y<10\}

• \{P2\} S \{Q2\}  \{x>10\} S \{y<10\}
Backward Reasoning

• Given S and Q, find P such that \( \{P\} S \{Q\} \).
• But which P?
  – \( \{x > 0\} \ y = x\times x \ \{y > 0\} \)
  – \( \{x \neq 0\} \ y = x\times x \ \{y > 0\} \)

• Weakest precondition P:
  – \( \{P\} S \{Q\} \)
  – For all P2, if \( \{P2\} S \{Q\} \) then P2 \( \Rightarrow \) P.
• Most relaxed requirements on program state.

\[ wp(S;S2, Q) \equiv wp(S1;wp(S2,Q)) \]

\[ wp(y=x+1; \ z=y+1; \ z > 2) \]

\[ \equiv wp(y=x+1, wp(z=y+1, z > 2)) \]
\[ \equiv wp(y=x+1, y+1 > 2) \]
\[ \equiv x+1+1 > 2 \]
\[ \equiv x > 0 \]

\[ wp(x = e, Q) \equiv Q[x := e] \]

\[ wp(x = y*y, x > 4) \equiv y*y > 4 \]
\[ \equiv |y| > 2 \]

\[ wp(if \ b \ S1 \ else \ S2, Q) \equiv (b \ \land \ wp(S1,Q)) \]
\[ \lor (\neg b \ \land \ wp(S2,Q)) \]

S:
\[ if \ (x < 5) \{ \]
\[ \quad x = x\times x; \]
\[ \quad wp(S, x \geq 9) \equiv \]
\[ \quad (x<5 \ \land \ wp(x=x\times x, x\geq9)) \]
\[ \} \]
\[ else \{ \]
\[ \quad x = x+1; \]
\[ \quad \lor (x\geq5 \ \land \ wp(x=x+1, x\geq9)) \equiv \]
\[ \quad (x<5 \ \land \ x*x>9) \lor (x\geq5 \ \land \ x+1\geq9) \equiv \]
\[ \{x \geq 9\} \]
\[ \quad (x<5 \ \land \ |x|\geq3) \lor (x\geq5 \ \land \ x\geq8) \]