CS 326
Formal Reasoning
Stephen Freund

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 \( n \),
    \( n.next == nil \lor n.next.prev == n \)
  – ...

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 \land z = 0
if (x != 0) {
  // x >= 0 \land z = 0 \land x != 0 (so x > 0)
  z = x;
  // ... \land z > 0
} else {
  // x >= 0 \land z = 0 \land !(x != 0) (so x = 0)
  z = x + 1;
  // ... \land z = 1
}
// (... \land z > 0) \lor (... \land z = 1) (so z > 0)

Hoare Triples

\hspace{1cm} \{ 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

\[
\begin{align*}
{x \neq 0} & \quad y = x \cdot x; \quad \{y > 0\} \\
{z \neq 1} & \quad y = z \cdot z; \quad \{y \neq z\} \\
{x \geq 0} & \quad y = 2 \cdot x; \quad \{y > x\} \\
\text{(true) if } (x > 7) \{ y=4; \text{ else } y=3; \} \{y < 5\} \\
{\text{(true)} x = y; \quad z = x; \quad \{y=z\} \\
{x=7 \land y=5} & \quad \text{tmp=}x; \quad x=\text{tmp}; \quad y=x; \quad \{y=7 \land x=5\}
\end{align*}
\]

**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:
  \( z \geq 1 \)
  
  \[
  \begin{align*}
  y &= z + 1; \quad \{w>y\} \\
  w &= y \cdot y; \quad \{w>y\}
  \end{align*}
  \]

**Conditional**

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

- Valid if: there are \( Q1, Q2 \) such that:
  1. \( \{ P / \ b \} S1 \{ Q1 \} \)
  2. \( \{ P / \ !b \} S2 \{ Q2 \} \)
  3. \( Q1 \lor Q2 \) implies \( Q \)

- Example:
  \( \{ \text{true} \} \)
  
  \[
  \begin{align*}
  \text{if } (x > 7) & \quad y = x; \quad \{ Q1 \} \\
  & \quad y = 20; \quad \{ Q2 \} \\
  \text{else } & \quad y = 20; \quad \{ y > 5 \}
  \end{align*}
  \]
Conditional

```c
{true}
if (x > 7)
    {true \land x>7}
    y = x;
    {y>7}
else
    {true \land x\leq 7}
    y = 20;
    {y=20}
{y>5}
```

Weaker vs. Stronger Assertions

• If $P_1 \Rightarrow P_2$ then:
  – $P_1$ is stronger than $P_2$
  – $P_2$ is weaker than $P_1$

• $x > 0 \Rightarrow x \geq 0$?
• $x = 1 \Rightarrow x > 0$?
• $x > 0 \Rightarrow true$?
• $x > 0 \Rightarrow x \neq 1$?

Weaker vs. Stronger Assertions

• Whenever $P_1$ holds, $P_2$ is guaranteed to hold

Strength and Hoare Logic

Suppose $\{P_1\} S \{Q_1\}$.  $\{x>0\} S \{y<5\}$
Strength and Hoare Logic

Suppose \{P1\} S \{Q1\}.

- \{P2\} S \{Q1\} \{x>10\} S \{y<5\}
- \{P1\} S \{Q2\} \{x>0\} S \{y<10\}

Strength and Hoare Logic

Suppose \{P1\} S \{Q1\}.

- \{P2\} S \{Q1\} \{x>10\} S \{y<5\}
- \{P1\} S \{Q2\} \{x>0\} S \{y<10\}

Strength and Hoare Logic

Suppose \{P1\} S \{Q1\}.

- \{P2\} S \{Q1\} \{x>10\} S \{y<5\}
- \{P1\} S \{Q2\} \{x>0\} S \{y<10\}

Strength and Hoare Logic

Suppose \{P1\} S \{Q1\}.

- \{P2\} S \{Q1\} \{x>10\} S \{y<5\}
- \{P1\} S \{Q2\} \{x>0\} S \{y<10\}

Backward Reasoning

- Given S and Q, find P such that \{P\} S \{Q\}.
- But which P?
  - \{x>0\} y=x*x \{y>0\}
  - \{x!=0\} y=x*x \{y>0\}
- Weakest precondition P:
  - \{P\} S \{Q\}
  - For all P2, if \{P2\} S \{Q\} then P2 => P.
- Most relaxed requirements on program state.
Example: What is the weakest precondition \( P \) that ensures \( c \) is not null?

\{ P \}
Point \( c = \text{null} \);
int \( z \);
if (\( y < 0 \)) {
  \( z = -2 \times y \);
} else {
  \( z = x \);
}
if (\( z > 10 \)) {
  c = \text{new Point}(z, y);
}
\{ Q: c != \text{null} \}

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

\[
\text{wp}(x = y \times y, x > 4) \\
\equiv y \times y > 4 \\
\equiv |y| > 2
\]

\[
\text{wp}(S1; S2, Q) \equiv \text{wp}(S1, \text{wp}(S2, Q))
\]

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

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

\[
S:
\text{if} (x < 5) \{ \\
  x = x \times x; \quad \text{wp}(S, x \geq 9) \equiv \quad \text{(x<5} \land \text{wp(x=x\times x, x\geq9))} \\
  \text{wp}(S, x \geq 9) \equiv \quad \text{(x\geq5} \land \text{wp(x=x+1, x\geq9))} \\
  \}
\text{else} \{ \\
  x = x + 1; \quad \text{wp}(S, x \geq 9) \equiv \quad \text{(x<5} \land \text{wp(x=x+1, x\geq9))} \\
  (x<5 \land x \times x > 9) \lor (x\geq5 \land x+1 \geq 9) \equiv \quad \text{(x<5} \land |x| \geq 3) \lor (x\geq5 \land x \geq 8) \\
  \}
\{ x \geq 9 \} \equiv \quad \text{(x\geq5} \land x \geq 9)
\]