Hoare Logic

CS 81
September 30, 2010
State Transformers

State before

\[ x = x + 1 \]

State after

\[ x = x + 1 \]

- \( x = 1 \)
- \( x = 7 \)
- \( \text{even}(x) \)
- \( x > 3 \)

precondition

- \( x = 2 \)
- \( x = 8 \)
- \( \text{odd}(x) \)
- \( x > 4 \)

postcondition
Preconditions and Postconditions

\[ x = x + 1 \]

**precondition**

- \( x == 1 \)
- \( x > 10 \)
- \( \top \)
- \( \bot \)

**postcondition**
Preconditions and Postconditions

precondition \quad x = x + 1

postcondition

x == 1
x > 10
\top
\bot
Design by Contract

precondition

postcondition

implementation
Specification for Sorting?

precondition

int a[N];

postcondition

∀i. (1 ≤ i < N) → a[i-1] ≤ a[i]

implementation

for (int i=0; i<n; ++i)
  a[i] = 0;
Hoare Triples

\{ precondition \} \quad code \quad \{ postcondition \}
Compare

\{ P \} \subset \{ Q \}

\{ P \} \subset \{ Q' \}

\{ P' \} \subset \{ Q \}

Assume P \rightarrow P' and Q \rightarrow Q'
First Rule of Hoare Logic

\[ P \rightarrow P' \quad \{ P' \} \; c \; \{ Q' \} \quad Q' \rightarrow Q \quad \text{IMPLIED} \]

\[ \{ P \} \; c \; \{ Q \} \]
Sequencing

\{ P_1 \} C_1 \{ Q_1 \}

\{ P_2 \} C_2 \{ Q_2 \}

\{ ??? \} C_1; C_2 \{ ??? \}
Second Rule of Hoare Logic

\[
\begin{array}{c}
\{ P \} \ c_1 \ \{ R \} \\
\{ P \} \ c_1; c_2 \ \{ Q \}
\end{array}
\quad \rightarrow 
\begin{array}{c}
\{ R \} \ c_2 \ \{ Q \}
\end{array}
\quad \text{COMPOSITION}
\]
What if Our Conditions Don’t Match?

\{ \top \} \ C_1 \ \{ x > 7 \}

\{ x > 5 \} \ C_2 \ \{ y = 2 \}

\{ ??? \} \ C_1 \ ; \ C_2 \ \{ ??? \}
If Statement

\[
\begin{align*}
\{ P \land e \} & \quad c_1 \quad \{ Q \} \quad \{ P \land \neg e \} & \quad c_2 \quad \{ Q \} \\
\{ P \} & \quad \text{if (e)} \quad c_1 \quad \text{else} \quad c_2 \quad \{ Q \} 
\end{align*}
\]
While Statement

\[
\begin{align*}
\{I \land e\} & \quad \text{c} \quad \{I\} \\
\{I\} & \quad \text{while } (e) \quad \text{c} \quad \{I \land \neg e\} \\
\end{align*}
\]
Assignments

\[
\{ P[e/x] \} \quad x = e; \quad \{ P \}
\]
Huth & Ryan: “Proof Tableaux”

\{ P_1 \}
C_1
\{ P_2 \}
C_2
\{ P_3 \}
...
\{ P_n \}
C_n
\{ P_{n+1} \}

Hint
Work Bottom Up!
Example

\{ y = 5 \} \ y = y + 1 \{ y = 6 \}

\{ y = 5 \}
\{ y + 1 = 6 \} \quad \text{implied}
\ y = y + 1
\{ y = 6 \} \quad \text{assignment}
Exercise: Swap

\{ x = x_0 \land y = y_0 \} \; t = x; \; x = y; \; y = t \; \{ x = y_0 \land y = x_0 \}
Exercise: Max

\{ \top \} \text{ if } (x > y) \text{ then } m = x \text{ else } m = y \{ m = \max(x,y) \}
Exercise: While

\[ \{ x \leq n \} \text{ while } (x < n) \ x = x+1 \ \{ x = n \} \]
Note: *Partial Correctness*

\[
\{ \top \} \\
\text{while (true) \{} \\
\quad x = x+1; \\
\text{\}} \\
\{ y = 99 \} 
\]
Total Correctness

= 

Partial Correctness

+ 

Termination
Proving Termination

One approach: Define a variant

(non-negative but decreases on each iteration)
Total Correctness?

```java
x = 0;
while ( n > 0 )
{
    x = x+1;
    n = n-1;
}
```
Termination?

n = 0;
while ( n < 99 )
{
    x = x+1;
    n = n+1;
}
Total Correctness?

\[
\{ m = m_0 > 0 \land n = n_0 > 0 \} \\
\]

while ( m != n ) {
   if ( m < n )
      n = n - m;
   else
      m = m - n;
}

\[
\{ m = \text{gcd}(m_0, n_0) \} \\
\]
Total Correctness?

\[
\{ x = 0 \land y = 1 \land z = 1 \land n \geq 1 \} \\
\text{while } ( z < n ) \{ \\
\hspace{1em} y = x + y; \\
\hspace{1em} x = y - x; \\
\hspace{1em} z = z + 1; \\
\} \\
\{ y = \text{fib}(n) \}
\]