Hoare Logic

CS 81: Computability and Logic
October 1, 2012
State Transformers

State before

\[ x = x + 1 \]

State after

\[ x = x + 1 \]

x==1

x==7

even(x)

x > 3

precondition

x==2

x==8

odd(x)

x > 4

postcondition
Preconditions and Postconditions

precondition

\[ x = x + 1 \]

postcondition

\[ x == 1 \]
\[ x > 10 \]
\[ \top \]
\[ \bot \]
Preconditions and Postconditions

precondition $x = x + 1$

postcondition $x == 1$

$\top$

$\bot$

$x > 10$

$\top$

$\bot$
Design by Contract

precondition → implementation → postcondition
Specification for Sorting?

precondition

```plaintext
int a[N];
```

postcondition

```plaintext
∀i. (1 ≤ i < N) → a[i-1] ≤ a[i]
```

implementation

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

{ precondition }  code  { postcondition }
Weakest Preconditions?

\{ ??? \} \ x = x + 2 \quad \{ x > 7 \}

\{ ??? \} \ x = 2 \quad \{ xy > 7 \}

\{ ??? \} \ x = y + z \quad \{ xy > 7 \}

\{ ??? \} \ x = e \quad \{ Q(x) \}
Assignments

\[
\begin{align*}
\{ P[e/x] \} & \quad x = e; \quad \{ P \} \\
\hline
\end{align*}
\]

ASSIGNMENT
But...

Are these provable? Are they “true”?

\[
\begin{align*}
\{ 49 > 42 \} & \quad x = 7 & \{ x^2 > 42 \} \\
\{ \top \} & \quad x = 7 & \{ x^2 > 42 \} \\
\{ 9 < y \} & \quad x = 7 & \{ y > x + 2 \} \\
\{ y > 9 \} & \quad x = 7 & \{ y > x \}
\end{align*}
\]

\[
\begin{array}{c}
\{ P[e/x] \} \\
\hline
x = e; \\
\{ P \}
\end{array}
\]

Assignment
Applying “Normal” Logic

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

\[ \{ P \} c \{ Q \} \]
Weakest Precondition?

\{ ??? \} x = 3; y = 4 \{ yz > 7x \}

\underbrace{\{ P[e/x] \}} x = e; \underbrace{\{ P \}} \text{ ASSIGNMENT}
Sequential Code

\[
\{ P \} \ c_1 \ \{ R \} \ \frac{\{ R \} \ c_2 \ \{ Q \} \ \text{COMPOSITION}}{\{ P \} \ c_1; c_2 \ \{ Q \}}
\]
What if Our Conditions Don’t Match?

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

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

\{ ??? \} C_1; C_2 \{ ??? \}
Huth & Ryan Proof Format (Sequential Code)

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

Hint
Work Bottom Up!
Huth & Ryan Proof Format (Sequential Code + Implies)

\[
\begin{align*}
\{ P_{1a} \} \\
\{ P_{1b} \} \\
C_1 \\
\{ P_{2a} \} \\
C_2 \\
\{ P_3 \} \\
\vdots \\
\{ P_n \} \\
C_n \\
\{ P_{n+1} \}
\end{align*}
\]

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}
Sequencing Proof

\[ \{ 4z > 21 \} \quad x = 3; \quad y = 4 \quad \{ yz > 7x \} \]

\[ \{ 4z > 21 \} \]
\[ \{ 4z > 7 \times 3 \} \]
\[ x = 3 \]
\[ \{ 4z > 7x \} \]
\[ y = 4 \]
\[ \{ yz > 7x \} \]
Exercise: Swap

\{ x = x_0 \land y = y_0 \} \quad t = x; \quad x = y; \quad y = t \{ x = y_0 \land y = x_0 \}
If Statements

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

To get from here to here

Need to complete two sub-proofs
Exercise: Max

\{ T \}
if (x > y):
    m = x
else:
    m = y
\{ m = \text{max}(x,y) \}
Invariants

- Conditions that are “preserved” by code
  - Both a precondition and a postcondition

- Loop invariant:
  True before and after every iteration of a loop.

- Representation Invariant
  True before and after every (public) function on some data structure.
While Statement

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

To get from here to here

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

Need to complete a sub-proof
Invariant?

\[
\begin{align*}
& \{ x = 0 \land y = 1 \land z = 1 \land n \geq 1 \} \\
\text{while ( } z < n \text{ ):} \\
& \hspace{1em} y = x + y \\
& \hspace{1em} x = y - x \\
& \hspace{1em} z = z + 1 \\
& \{ y = \text{fib}(n) \} 
\end{align*}
\]
Invariant?

\{ \ m = m_0 > 0 \wedge n = n_0 > 0 \ \}

while ( m != n ):
    if ( m < n ):
        n = n-m
    else:
        m = m-n

\{ \ m = \gcd(m_0, n_0) \ \}
Exercise: While

\{ x \leq n \} \text{ while } (x < n) \ x = x + 1 \ \{ x = n \}
Partial Correctness

\{ T \}
while (true):
    x = x+1
\{ y = 99 \}

\{ y = 4 \}
while (y != 42):
    b = not b
\{ y = 42 \}
Total Correctness = Partial Correctness + Termination
Proving Termination

One approach: Define a variant

(non-negative but decreases on each iteration)
n = 0
while ( n < 99 ):
    x = x+1
    n = n+1

Termination?
Termination?

\[
\begin{align*}
\{ \ n = n_0 > 0 \} \\
\text{x = 0;}
\end{align*}
\]

while ( n > 0 ):
  \[
  \begin{align*}
  \text{x = x+1;}
  \text{n = n-1;}
  \end{align*}
\]

\[
\{ \ x = n_0 \}
\]
Partial Correctness?

\{ n = n_0 > 0 \}
x = 0;
while ( n > 0 ):
x = x+1;
n = n - 1;
\{ x = n_0 \}

\{ n = n_0 > 0 \}
x = 0;
\{ I \}
while ( n > 0 ):
\{ I \}
x = x+1;
\{ ... \}
n = n - 1;
\{ I \}
\{ I \land n \leq 0 \}
\{ x = n_0 \}
Total Correctness?

\[
\{ m = m_0 > 0 \land n = n_0 > 0 \} \\
\text{while ( } m \neq n \text{ ):} \\
\quad \text{if ( } m < n \text{ )}: \\
\quad\quad n = n - m \\
\quad\text{else:} \\
\quad\quad m = m - n \\
\{ m = \gcd(m_0, n_0) \} \\
\]

\[
\{ m = m_0 > 0 \land n = n_0 > 0 \} \\
\{ I \} \\
\text{while ( } m \neq n \text{ ):} \\
\quad \{ I \} \\
\quad \text{if ( } m < n \text{ ):} \\
\quad\quad \{ I \land m < n \} \\
\quad\quad n = n - m \\
\quad\text{else:} \\
\quad\quad \{ I \land m \geq n \} \\
\quad\quad m = m - n \\
\quad \{ I \} \\
\quad \{ I \land m = n \} \\
\{ m = \gcd(m_0, n_0) \} \\
\]
Total Correctness?

\[
\{ x = 0 \land y = 1 \land z = 1 \land n \geq 1 \} \\

\text{while ( } z < n \text{ ):} \\
\text{\quad } y = x + y \\
\text{\quad } x = y - x \\
\text{\quad } z = z + 1 \\
\{ y = \text{fib}(n) \}