

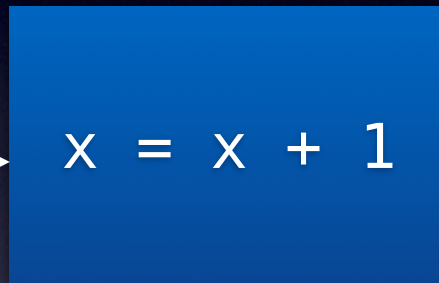
# Hoare Logic

CS 81

September 30, 2010

# State Transformers

State before



State after

$x == 1$

$x == 7$

$\text{even}(x)$

$x > 3$

*precondition*

$x == 2$

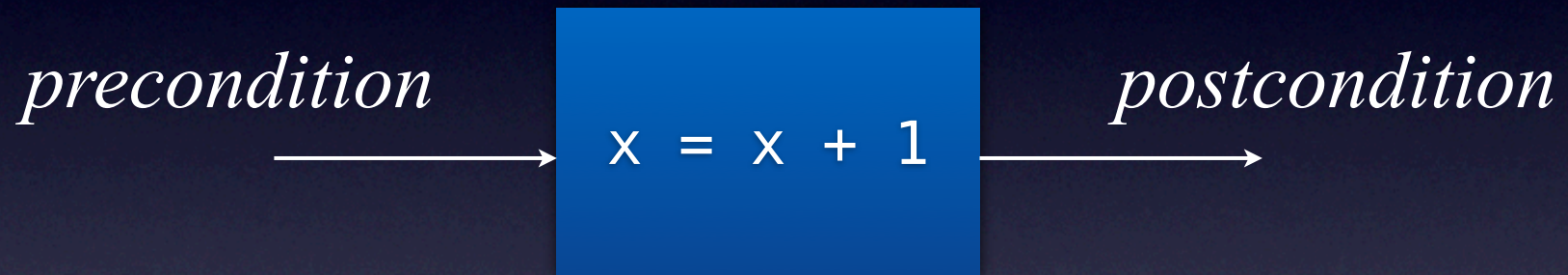
$x == 8$

$\text{odd}(x)$

$x > 4$

*postcondition*

# Preconditions and Postconditions



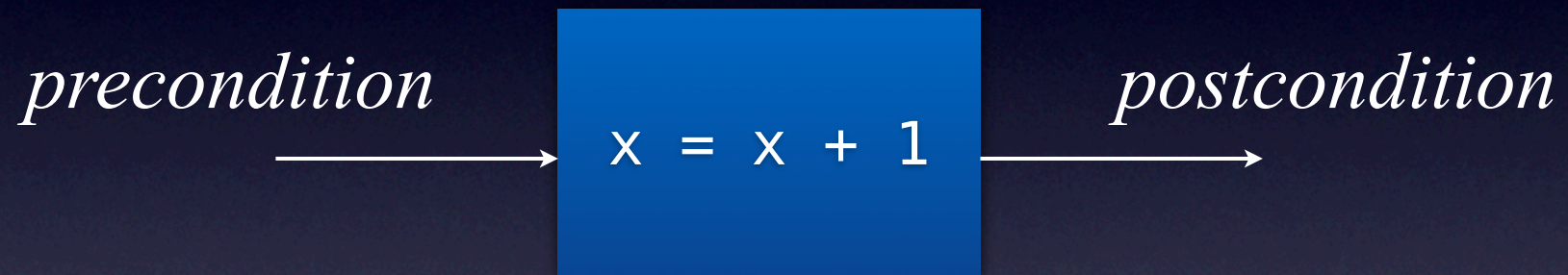
`x == 1`

`x > 10`

⊤

⊥

# Preconditions and Postconditions



$x == 1$

$x > 10$

$\top$

$\perp$

# Design by Contract

*precondition*

*postcondition*

*implementation*



# Specification for Sorting?

*precondition*

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

*postcondition*

$$\forall i. (1 \leq i < N) \rightarrow a[i-1] \leq a[i]$$

*implementation*

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

# Hoare Triples



$\{ \textit{precondition} \} \quad \text{code} \quad \{ \textit{postcondition} \}$

# Compare

$$\{P\} \text{ c } \{Q\}$$

$$\{P\} \text{ c } \{Q'\}$$

$$\{P'\} \text{ c } \{Q\}$$

Assume  $P \rightarrow P'$  and  $Q \rightarrow Q'$

# First Rule of Hoare Logic

$$\frac{P \rightarrow P' \quad \{ P' \} c \{ Q' \} \quad Q' \rightarrow Q}{\{ P \} c \{ Q \}} \text{ IMPLIED}$$

# Sequencing

$\{ P_1 \} C_1 \{ Q_1 \}$

$\{ P_2 \} C_2 \{ Q_2 \}$

$\{ ??? \} C_1 ; C_2 \{ ??? \}$

# Second Rule of Hoare Logic

$$\frac{\{ P \} c_1 \{ R \} \quad \{ R \} c_2 \{ Q \}}{\{ P \} c_1; c_2 \{ Q \}} \text{COMPOSITION}$$

# What if Our Conditions Don't Match?

$$\{ \top \} C_1 \{ x > 7 \}$$
$$\{ x > 5 \} C_2 \{ y = 2 \}$$
$$\{ ??? \} C_1; C_2 \{ ??? \}$$

# If Statement

$$\frac{\{ P \wedge e \} c_1 \{ Q \} \quad \{ P \wedge \neg e \} c_2 \{ Q \}}{\{ P \} \text{ if } (e) c_1 \text{ else } c_2 \{ Q \}} \text{ IF}$$

# While Statement

$$\frac{\{ I \wedge e \} c \{ I \}}{\{ I \} \text{ while } (e) c \{ I \wedge \neg e \}} \text{ WHILE}$$

# Assignments

$$\frac{\{ P[e/x] \}}{\{ P \}} \quad x = e; \quad \text{ASSIGNMENT}$$

# 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

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

$$\begin{array}{ll} \{ y = 5 \} & \\ \{ y + 1 = 6 \} & \text{implied} \\ y = y + 1 & \\ \{ y = 6 \} & \text{assignment} \end{array}$$

# Exercise: Swap

$\{ x = x_0 \wedge y = y_0 \} \quad t = x; \quad x = y; \quad y = t \quad \{ x = y_0 \wedge y = x_0 \}$

# Exercise: Max

```
{ T } if (x > y) m = x else m = y { m = max(x,y) }
```

# Exercise: While

$\{x \leq n\}$  while  $(x < n)$   $x = x+1$   $\{x = n\}$

# Note: *Partial* Correctness

{  $\top$  }

```
while (true) {  
    x = x+1;  
}
```

{  $y = 99$  }

# Total Correctness

=

## Partial Correctness

+

## Termination

# Proving Termination

One approach: Define a *variant*

(non-negative but decreases on each iteration)

# Total Correctness?

```
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 = m0 > 0 ∧ n = n0 > 0 }
```

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

```
{ m = gcd(m0, n0) }
```

# Total Correctness?

```
{ x = 0 ∧ y = 1 ∧ z = 1 ∧ n ≥ 1 }
```

```
while ( z < n ) {  
  y = x + y;  
  x = y - x;  
  z = z + 1;  
}
```

```
{ y = fib(n) }
```