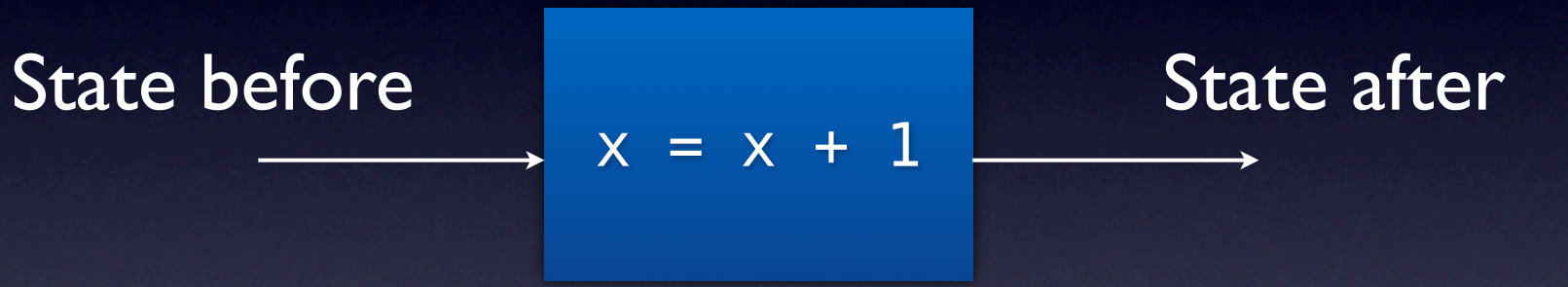
Hoare Logic

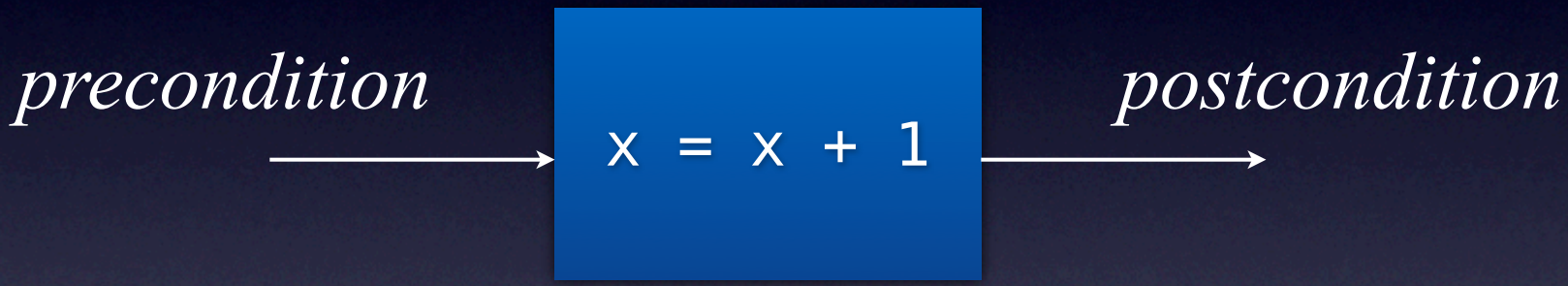
CS 81: Computability and Logic
October 5, 2011

State Transformers



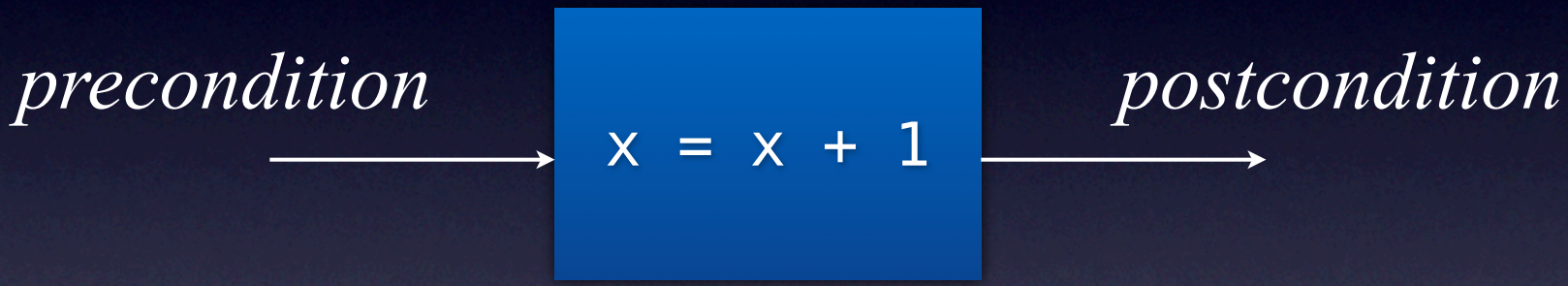
x==1	x==2
x==7	x==8
even(x)	odd(x)
x > 3	x > 4
precondition	*postcondition*

Preconditions and Postconditions




```
x == 1
x > 10
   ⊤
   ⊥
```

Preconditions and Postconditions




```
x == 1
x > 10
```

$\top$

$\bot$

Design by Contract

Specification for Sorting?

precondition

int a[N];

postcondition

$\forall i.\ (1 \le i < \texttt{N}) \rightarrow \texttt{a}[i\text{-}1] \le \texttt{a}[i]$

implementation

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

Hoare Triples

{ *precondition* } `code` { *postcondition* }

Weakest Preconditions?

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

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

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

Assignments

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

But...

Are these derivable? Are they "true"?

$\{ y > 9 \}\ x = 7\ \{ x^2 > 42 \}$

$\{ 9 < y \}\ x = 7\ \{ y > x + 2 \}$

$\{ y > 9 \}\ x = 7\ \{ y > x \}$

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

Applying "Normal" Logic

$$\frac{P \to P' \qquad \{\, P' \,\}\, c\, \{\, Q' \,\} \qquad Q' \to Q}{\{\, P \,\}\, c\, \{\, Q \,\}}\ \text{IMPLIED}$$

Weakest Precondition?

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

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

Sequential Code

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

What if Our Conditions Don't Match?

$$\{ \top \}\ \mathtt{c}_1\ \{ x > 7 \}$$

$$\{ x > 5 \}\ \mathtt{c}_2\ \{ y = 2 \}$$

$$\{ ??? \}\ \mathtt{c}_1 ; \mathtt{c}_2\ \{ ??? \}$$

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)

$\{ P_{1a} \}$
$\{ P_{1b} \}$
c_1
$\{ P_{2a} \}$
c_2
$\{ P_3 \}$
...
$\{ P_n \}$
c_n
$\{ P_{n+1} \}$

Hint
Work Bottom Up!

Example

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

$\{ y = 5 \}$

$\{ y + 1 = 6 \}$ implied

`y = y + 1`

$\{ y = 6 \}$ assignment

Sequencing Proof

$\{ 4z > 21 \}$ $x = 3;$ $y = 4$ $\{ yz > 7x \}$

$\{ 4z > 21 \}$
$\{ 4z > 7 \times 3 \}$
$x = 3$
$\{ 4z > 7x \}$
$y = 4$
$\{ yz > 7x \}$

Exercise: Swap

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

If Statements

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

To get from here to here

```
{P}
if (e):
    {P ∧ e}
    c1
    {Q}
else:
    {P ∧ ¬e}
    c2
    {Q}
{Q}
```

Need to complete two sub-proofs

Exercise: Max

{ ⊤ }

```
if (x > y):
    m = x
else:
    m = y
```

{ m = 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

$$\frac{\{\, I \wedge e \,\}\ c\ \{\, I \,\}}{\{\, I \,\}\ \texttt{while}\ (e)\texttt{:}\, c\ \{\, I \wedge \neg e \,\}}\ \textsc{while}$$

To get from here to here

Need to complete a sub-proof

Invariant?

$\{ x = 0 \wedge y = 1 \wedge z = 1 \wedge n \geq 1 \}$

```
while ( z < n ):
   y = x + y
   x = y - x
   z = z + 1
```

$\{ y = \text{fib}(n) \}$

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

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

Note: *Partial* Correctness

$\{ \top \}$

```
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)

Termination?

```
n = 0
while ( n < 99 ):
    x = x+1
    n = n+1
```

Termination?

$\{ n = n_0 > 0 \}$

```
x = 0;
while ( n > 0 ):
      x = x+1;
      n = n-1;
```

$\{ x = n_0 \}$

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;
```
$\{ \ldots \}$
```
      n = n-1;
```
$\{ I \}$

$\{ I \wedge n \leq 0 \}$

$\{ x = n_0 \}$

Total Correctness?

$\{ \mathtt{m} = \mathtt{m}_0 > 0 \wedge \mathtt{n} = \mathtt{n}_0 > 0 \}$

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

$\{ \mathtt{m} = \gcd(\mathtt{m}_0, \mathtt{n}_0) \}$

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

$\{ \mathtt{m} = \mathtt{m}_0 > 0 \wedge \mathtt{n} = \mathtt{n}_0 > 0 \}$
$\{ I \}$
while (m != n):
  $\{ I \}$
  if (m < n):
    $\{ I \wedge \mathtt{m} < \mathtt{n} \}$
    n = n-m
    $\{ I \}$
  else:
    $\{ I \wedge \mathtt{m} \geq \mathtt{n} \}$
    m = m-n
    $\{ I \}$
  $\{ I \}$
$\{ I \wedge \mathtt{m} = \mathtt{n} \}$
$\{ \mathtt{m} = \gcd(\mathtt{m}_0, \mathtt{n}_0) \}$

Total Correctness?

$\{ x = 0 \wedge y = 1 \wedge z = 1 \wedge n \geq 1 \}$

```
while ( z < n ):
   y = x + y
   x = y - x
   z = z + 1
```

$\{ y = fib(n) \}$