Program Verification

CMPSC 461
Programming Language Concepts
Penn State University
Fall 2016

```
{true}
x := 5;
y := 1;
{(y=1) ∧ (x=5)}
```

$$\text{sp}(\texttt{x:=5;y:=1}, \text{true}) = \text{sp}(\texttt{y:=1}, \text{sp}(\texttt{x:=5}, \text{true}))$$
$$= \text{sp}(\texttt{y:=1}, x = 5)$$
$$= \left(\exists v.\, (y = 1) \wedge (x = 5)\right)$$
$$= (y = 1) \wedge (x = 5)$$

The existential quantifier complicates the formula …

Goal: check if $\{P\}s\{Q\}$ is valid
Method 1: check $\text{sp}(s, P) \Rightarrow Q$

```
{true}
x := 5;
y := 1;
{(y=1) ∧ (x=5)}
```

$\text{sp}(\texttt{x:=5;y:=1}, \text{true}) = \text{sp}(\texttt{y:=1}, \text{sp}(\texttt{x:=5}, \text{true}))$
$= \text{sp}(\texttt{y:=1}, x = 5)$
$= (y = 1) \wedge (x = 5)$

The reasoning:

$\{\text{true}\}\texttt{x:=5}\ \{x = 5\}\ \texttt{y:=5}\ \{x = 5 \wedge y = 1\}$

sp computation (forward): →

Backward? ←

Goal: check if $\{P\}s\{Q\}$ is valid
Method 1: check $\text{sp}(s, P) \Rightarrow Q$

Weakest Precondition

Goal: check if $\{P\}s\{Q\}$ is valid
Method 1: check $\mathrm{sp}(s, P) \Rightarrow Q$
Method 2: check $P \Rightarrow \mathrm{wp}(s, Q)$

$\mathrm{wp}(s, Q)$ is the **weakest precondition** of s, w.r.t. Q
Property: $\{P\}s\{Q\}$ is valid iff $P \Rightarrow \mathrm{wp}(s, Q)$

Hence, validity of a triple $\{P\}s\{Q\}$ is equivalent to the truth value of proposition $P \Rightarrow \mathrm{wp}(s, Q)$

Assignment Rule (Hoare's Axiom)

$$\boxed{\text{wp}(\texttt{x:=e}, Q) = Q[x \leftarrow e]}$$

Examples:

wp($\texttt{x:=5}$, $x = 5$) = $(5 = 5)$ = (true)

wp($\texttt{x:=x+3}$, $x = y + 3$) = $(x + 3 = y + 3)$

= $(x = y)$

This rule is simpler than Floyd's axiom, hence weakest precondition is used in most systems

Composition Rule

$$\text{wp}(\texttt{s1;s2}, Q) = \text{wp}(\texttt{s1}, \text{wp}(\texttt{s2}, Q))$$

```
{true}
x := 5;
y := 1;
{(y=1) ∧ (x=5)}
```

$\text{wp}(\texttt{x:=5;y:=1}, (x = 5) \wedge (y = 1))$

$= \text{wp}(\texttt{x:=5}, \text{wp}(\texttt{y:=1}, (x = 5) \wedge (y = 1)))$

$= \text{wp}(\texttt{x:=5}, (x = 5) \wedge (1 = 1))$

$= (5 = 5) \wedge (1 = 1)$

$= \text{true}$

Composition Rule

$$\mathrm{wp}(\texttt{s1;s2}, Q) = \mathrm{wp}(\texttt{s1}, \mathrm{wp}(\texttt{s2}, Q))$$

```
{true}
x := 5;
x := 2;
{x=2}
```

$\mathrm{wp}(\texttt{x:=5;x:=2}, x = 2)$

$= \mathrm{wp}(\texttt{x:=5}, \mathrm{wp}(\texttt{x:=2}, x = 2))$

$= \mathrm{wp}(\texttt{x:=5}, 2 = 2)$

$= (2 = 2)$

$= \mathrm{true}$

Branch Rule

$$\boxed{\begin{aligned}&\text{wp}(\text{if}(E)\ \texttt{s1}\ \text{else}\ \texttt{s2}, Q) = \\ &\quad (E \Rightarrow \text{wp}(\texttt{s1}, Q) \land \neg E \Rightarrow \text{wp}(\texttt{s2}, Q))\end{aligned}}$$

```
{true}
if (x>0)
  y := x;
else
  y := -x;
{y≥0}
```

$$\begin{aligned}&\text{wp}(\text{P}, y \geq 0)\\ &= x > 0 \Rightarrow \text{wp}(\texttt{y} := \text{x}, y \geq 0) \land\\ &\neg(x > 0) \Rightarrow \text{wp}(\texttt{y} := -\text{x}, y \geq 0))\\ &= (x > 0 \Rightarrow x \geq 0) \land (x \leq 0 \Rightarrow -x \geq 0)\\ &= \text{true}\end{aligned}$$

Program and Logics

We need to formally specify

1) The desired property

```
int Max(int a, int b) {
  int m;
  if (a>b)  m:=a;
  else      m:=b;
  return m;
}
```

Precondition (true)

Postcondition $(m = \max(a, b))$

Program and Logic

We need to formally specify

1) The desired property
2) The behavior of program

Hoare Triple: $\{P\}s\{Q\}$

A program s is correct w.r.t. P and Q iff $\{P\}s\{Q\}$ is valid

A triple is valid iff $P \Rightarrow \mathrm{wp}(s, Q)$ is true

Computing WP

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

$$\text{wp}(s_1;\ s_2, Q) = \text{wp}(s_1, \text{wp}(s_2, Q))$$

$$\text{wp}(\text{if}(E)\ s_1 \text{ else } s_2, Q) = (E \Rightarrow \text{wp}(s_1, Q) \wedge \neg E \Rightarrow \text{wp}(s_2, Q))$$

$$\text{wp}(\text{nop}, Q) = Q$$

A dummy operation that has no effects

Example

```
{x>0}
x := x+1;
y := x * (x+5);
{y>0}
```

Goal: show the Hoare triple is valid

1) Compute wp(prog, postcondition)

wp(x:=x+1; y:=x*(x+5), $y > 0$)

= wp (x:=x+1, wp (y:=x*(x+5), $y > 0$))

= wp (x:=x+1, $x * (x + 5) > 0$))

= $(x + 1) * (x + 6) > 0$

2) Show the precondition implies wp

$(\mathrm{x} > 0) \Rightarrow ((x + 1) * (x + 6) > 0)$

Example

```
{true}
int m;
if (a>b)  m:=a;
else      m:=b;
{m=max(a,b)}
```

Goal: show the Hoare triple is valid

1) Compute wp(prog, postcondition)

wp(if (a>b) m:=a else m:=b, $m = \max(a, b)$)

= $(a > b \Rightarrow$ wp(m:=a, $m = \max(a, b)$)) $\wedge$

$(a \leq b \Rightarrow$ wp(m:=b, $m = \max(a, b)$))

= $(a > b \Rightarrow a = \max(a, b)) \wedge (a \leq b \Rightarrow b = \max(a, b))$

= true

2) Show the precondition implies wp

$\text{true} \Rightarrow \text{true}$

Example

```
{x=a}
if (a<0) x := -a;
{x=|a|}
```

Goal: show the Hoare triple is valid

1) Compute wp(prog, postcondition)

wp(if (a<0) x:=-a ,$x = |a|$)

= ($a < 0 \Rightarrow$wp (x:= –a, $x = |a|$)) ∧

($a \geq 0 \Rightarrow$wp (nop, $x = |a|$))

= $(a < 0 \Rightarrow -a = |a|) \wedge (a \geq 0 \Rightarrow x = |a|)$

= $(a \geq 0 \Rightarrow x = |a|)$

2) Show the precondition implies wp

$(\mathrm{x} = a) \Rightarrow (a \geq 0 \Rightarrow x = |a|)$

Computing WP

$$\text{wp}(\texttt{x := e}, Q) = Q[x \leftarrow e]$$

$$\text{wp}(s_1;\ s_2, Q) = \text{wp}(s_1, \text{wp}(s_2, Q))$$

$$\text{wp}(\text{if}(E)\ s_1 \text{ else } s_2, Q) = (E \Rightarrow \text{wp}(s_1, Q) \wedge \neg E \Rightarrow \text{wp}(s_2, Q))$$

$$\text{wp}(\text{nop}, Q) = Q$$

Observation: program verification is systematic and automatic if there is no loop!