Program Verification

CMPSC 461
Programming Language Concepts
Penn State University
Fall 2016

Final

Cumulative, 35% of final grade

Dec. 14 (Wed.), 6:50-8:40PM, 119 Osmond Lab

Conflict exam:

Dec. 12 Mon. 6:50-8:40PM, 323 Boucke

(if you have officially registered for it via University)

This Week

HW6 is due this Friday at ***NOON***, no late submissions

Practice problems posted on Canvas

This Friday: review

Computing WP

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

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

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

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

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

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, $x = |a|$)) $\wedge$

($a \geq 0 \Rightarrow$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

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

Loops $\{P\}\text{while } (E)\ s\ \{Q\}$

What is the WP?

Let W= $\text{while } (E)\ s$, then $\{P\}\text{while } (E)\ s\ \{Q\}$

is the same as $\{P\}\text{if } (E)\ s; \text{W else nop } \{Q\}$

By if-rule,

$$\begin{aligned}\text{wp}(W, Q) &= (E \Rightarrow \text{wp}(s; W, Q) \wedge \neg E \Rightarrow Q) \\ &= (E \Rightarrow \text{wp}(s; \text{wp}(W, Q)) \wedge \neg E \Rightarrow Q)\end{aligned}$$

Loop Invariant

Loop Invariant $\{P\}\text{while } (E)\ s\ \{Q\}$

$$Inv = (E \Rightarrow \text{wp}(s, Inv) \land \neg E \Rightarrow Q)$$

Hence, $Inv \land E \Rightarrow \text{wp}(s, Inv)$ and $Inv \land \neg E \Rightarrow Q$

(Proof is beyond the scope of this lecture)

Loop invariant (Inv) is a proposition that is:

1) Initially true ($P \Rightarrow Inv$)
2) True after each iteration ($Inv \land E \Rightarrow \text{wp}(s, Inv)$)
3) Termination of loop implies the postcondition ($Inv \land \neg E \Rightarrow Q$)

Loop Invariant and Induction

Loop invariant (Inv) is a proposition that is:

1) Initially true ($P \Rightarrow Inv$)
2) True after each iteration ($Inv \land E \Rightarrow \mathrm{wp}(s, Inv)$)
3) Termination of loop implies the postcondition

   ($Inv \land \neg E \Rightarrow Q$)

Intuitively, we are proving the correctness of an arbitrary number of loop iterations, by **induction**!

Example

```
{n ≥ 0}
r:=0, i:=0;
while (i<n) {
  r := r+2;
  i ++;
}
{r = 2×n}
```

Goal: show the Hoare triple is valid

1) Write down a tentative loop invariant (Inv)

$r = 2 \times i \land i \leq n$

2) Show *Inv* is a loop invariant

- $\{n \geq 0\}$ r:=0, i=0; $\{Inv\}$ is valid
- $Inv \land i < n \Rightarrow$ wp(r:=r+2; i++, Inv)
- $Inv \land i \geq n \Rightarrow r = 2 \times n$

Example

```
{n ≥ 0}
r:=1, i:=n;
while (i>0) {
  r := r*i;
  i --;
}
{r = n!}
```

Goal: show the Hoare triple is valid

1) Write down a tentative loop invariant (Inv)

$r = \prod_{j=i+1}^{n} j \wedge i \geq 0 \wedge n \geq 0$

2) Show *Inv* is a loop invariant

- $\{n \geq 0\}$ r:=1, i=n; $\{Inv\}$ is valid
- $Inv \wedge i > 0 \Rightarrow$ wp(r:=r*i; i−−;, Inv)
- $Inv \wedge i \leq 0 \Rightarrow r = n!$

Verification in Practice

Goal: show the Hoare triple is valid

1) Write down a tentative loop invariant (Inv)
2) Show *Inv* is a loop invariant

Step 2) is automatic, but 1) is mostly manual …

Significant artifacts (e.g., simple OS) have been verified, but with pains (e.g., 3 person-years)

Total vs. Partial Correctness

$\{P\}\text{while }(E)\ s\ \{Q\}$

Partial correctness: if the loop terminates, Q must be true. However, the loop might not terminate

E.g., $\{P\}\text{while (true) } s\ \{Q\}, Inv \wedge \neg\text{true} \Rightarrow Q$ is true

Total correctness: prove loop determinates (undecidable in general)

Summary

Goal: prove a program s is correct

Step 1: formalize "correctness" by writing down the precondition P and postcondition Q

Step 2: show that the Hoare tripe ($\{P\}s\{Q\}$) is valid

- Mostly automatic, except for the loops

What is verified?

Given any state satisfying *P*, the final state after executing s must satisfy Q, if s terminates