



Proofs of Correctness: Introduction to Axiomatic Verification

- Introduction
- Weak correctness predicate
- Assignment statements
- Sequencing
- Selection statements
- Iteration

1



Introduction

- *What is Axiomatic Verification?*

A formal method of reasoning about the **functional** correctness of a **structured, sequential program** by tracing its state changes from an initial (i.e., pre-) condition to a final (i.e., post-) condition according to a set of self-evident rules (i.e., *axioms*).
- *What is its primary goal?*

To provide a means for “proving” (or “disproving”) the functional correctness of a sequential program with respect to its (formal) specification.

2



Introduction (cont.)

- *What are the benefits of studying axiomatic verification?*
 - Understanding its *limitations*.
 - **Deeper insights** into programming and program structures.
 - **Criteria for judging** both programs and programming languages.
 - The ability to formally verify small (or parts of large) sequential programs.

3



Weak Correctness Predicate

- To prove that program S is (weakly) correct with respect to pre-condition P and post-condition Q, it is sufficient to show: $\{P\} S \{Q\}$.
- Interpretation of $\{P\} S \{Q\}$: "**if** the input (initial state) satisfies the pre-condition P and (**if**) the program S executes and terminates, **then** the output (final state) will satisfy the post-condition Q."

4



Weak Correctness Predicate (cont.)

- Thus, $\{P\} S \{Q\}$ is *true* unless Q **could** be false if S terminates, given that P held before S executes.
- What are the truth values of the following assertions?
 - (1) $\{x=1\} y := x+1 \{y>0\}$
 - (2) $\{x>0\} x := x-1 \{x>0\}$

5



Weak Correctness Predicate (cont.)

- Thus, $\{P\} S \{Q\}$ is *true* unless Q **could** be false if S terminates, given that P held before S executes.
- What are the truth values of the following assertions?
 - (3) $\{1=2\} k := 5 \{k<0\}$

6



Weak Correctness Predicate (cont.)

- Thus, $\{P\} S \{Q\}$ is *true* unless Q **could** be *false* if S terminates, given that P held before S executes.
- What are the truth values of the following assertions?
(4) $\{\text{true}\} \text{while } x <> 5 \text{ do } x := x-1 \{x=5\}$
(Hint: When will S terminate?)

7



Weak Correctness Predicate (cont.)

- We now consider techniques for proving that such assertions hold for structured programs comprised of assignment statements, if-then (-else) statements, and while loops.

(Why these particular constructs?)

8

Reasoning about Assignment Statements

- For each of the following pre-conditions, P , and assignment statements, S , identify a “strong” post-condition, Q , such that $\{P\} S \{Q\}$ would hold.
- A “strong” post-condition captures all after-execution state information of interest.
- We ignore propositions such as $X=X'$ (“the final value of X is the same as the initial value of X ”).

9

Reasoning about Assignment Statements (cont.)

<u>{P}</u>	<u>S</u>	<u>{Q}</u>
{J=6}	K := 3	
{J=6}	J := J+2	
{A<B}	Min := A	
{X<0}	Y := -X	

10

Reasoning about Assignment Statements (cont.)

- For each of the following post-conditions, Q , and assignment statements, S , identify a “weak” pre-condition, P , such that $\{P\} S \{Q\}$ would hold.
(A “weak” pre-condition reflects only what **needs** to be true before.)

11

Reasoning about Assignment Statements (cont.)

<u>$\{P\}$</u>	<u>S</u>	<u>$\{Q\}$</u>
	$I := 4$	$\{J=7 \wedge I=4\}$
	$I := 4$	$\{I=4\}$
	$I := 4$	$\{I=17\}$
	$Y := X+3$	$\{Y=10\}$

12

When does
 $(\{P\} S \{Q\}) \Rightarrow (\{K\} S \{W\})?$

- We just determined that

$$\{J=7\} I := 4 \{J=7 \wedge I=4\}$$

holds.

- We can deduce from this that

$$\{J=7\} I := 4 \{J=7\}$$

also holds since $\{J=7 \wedge I=4\}$ is stronger than $\{J=7\}$, because

$$\{J=7 \wedge I=4\} \Rightarrow \{J=7\}.$$

13

When does
 $(\{P\} S \{Q\}) \Rightarrow (\{K\} S \{W\})?$

- Similarly, if we know that

$$\{J=7\} I := 4 \{J=7 \wedge I=4\}$$

holds, it follows that

$$\{J=7 \wedge K=17\} I := 4 \{J=7 \wedge I=4\}$$

also holds since $\{J=7\}$ is weaker than $\{J=7 \wedge K=17\}$, because

$$\{J=7 \wedge K=17\} \Rightarrow \{J=7\}.$$

14

When does

$(\{P\} S \{Q\}) \Rightarrow (\{K\} S \{W\})?$

- Thus, we can replace pre-conditions with ones that are *stronger*, and post-conditions with ones that are *weaker*.
- Note that if $A \Rightarrow B$, we say that A is *stronger* than B, or equivalently, that B is *weaker* than A.

15

Reasoning about Sequencing

- In general:
 - if you know $\{P\} S_1 \{R\}$ and
 - you know $\{R\} S_2 \{Q\}$
 - then you know $\{P\} S_1; S_2 \{Q\}$.

(So, to prove $\{P\} S_1; S_2 \{Q\}$, find $\{R\}$.)

16

Example 1

- Prove the assertion:

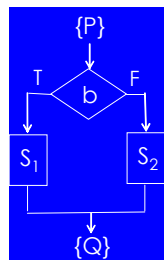
$\{A=5\} B := A+2; C := B-A; D := A-C \quad \{A=5 \wedge D=3\}$

17

Reasoning about If_then_else Statements

- Consider the assertion:

$\{P\} \text{ if } b \text{ then } S_1 \text{ else } S_2 \{Q\}$

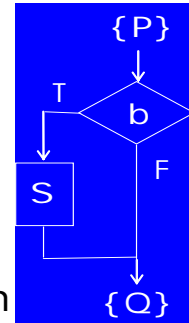


- What are the *necessary* conditions for this assertion to hold?

18

Reasoning about If_then Statements

- Consider the assertion:
 $\{P\} \text{ if } b \text{ then } S \{Q\}$
- What are the *necessary* conditions for this assertion hold?



19

Example 2

- Prove the assertion:

$\{Z=B\} \text{ if } A>B \text{ then } Z := A \{Z=\text{Max}(A,B)\}$

20



Proof Rules

- Before proceeding to while loops, let's capture our previous reasoning about sequencing, selection statements, and state condition replacement in appropriate *rules of inference*.

Rule for **Sequencing**:

$$\frac{\{P\} S_1 \{R\}, \{R\} S_2 \{Q\}}{\{P\} S_1; S_2 \{Q\}}$$

21



Proof Rules (cont.)

Rule for *if_then_else* statement:

$$\frac{\{P \wedge b\} S_1 \{Q\}, \{P \wedge \neg b\} S_2 \{Q\}}{\{P\} \text{if } b \text{ then } S_1 \text{ else } S_2 \{Q\}}$$

Rule for *if_then* statement:

$$\frac{\{P \wedge b\} S \{Q\}, (P \wedge \neg b) \Rightarrow Q}{\{P\} \text{if } b \text{ then } S \{Q\}}$$

22

Proof Rules (cont.)

Rule for **State Condition Replacement**:

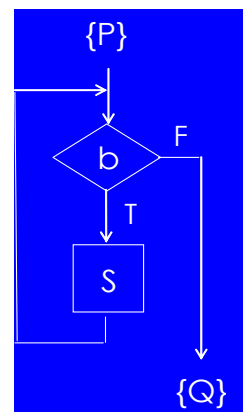
$$\frac{K \Rightarrow P, \{P\} S \{Q\}, Q \Rightarrow W}{\{K\} S \{W\}}$$

23

Reasoning about Iteration

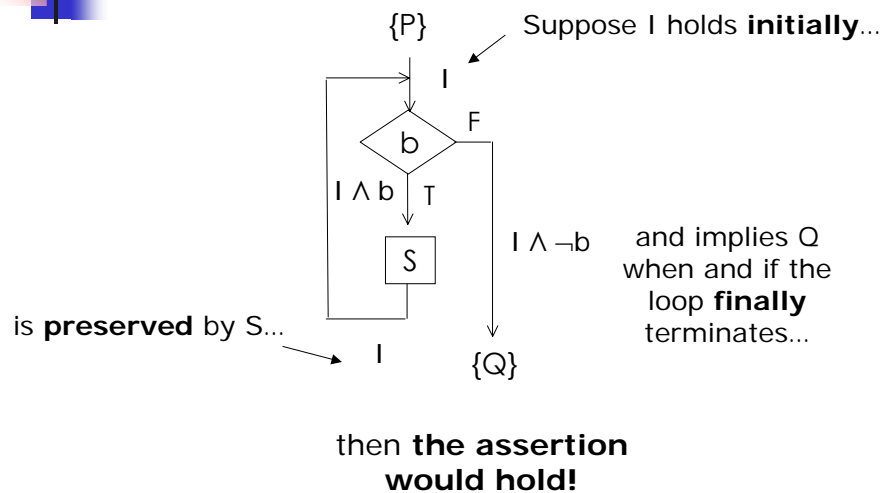
- Consider the assertion:
 $\{P\} \text{ while } b \text{ do } S \{Q\}$

What are the *necessary* conditions for this assertion to hold?



24

Consider a Loop "Invariant" - I



25

Sufficient Conditions: while_do

Thus, a Rule for the *while_do* statement is:

$$\frac{P \Rightarrow I, \{I \wedge b\} S \{I\}, (I \wedge \neg b) \Rightarrow Q}{\{P\} \text{ while } b \text{ do } S \{Q\}}$$

where the three antecedents are sometimes given the names *initialization*, *preservation*, and *finalization*, respectively.

26




Example 3

Use the invariant $I: Z=XJ$ to prove:

<pre> {true} Z := X J := 1 while J <> Y do Z := Z+X J := J+1 end_while {Z=XY} </pre>	<p><u>Initialization:</u> $P \Rightarrow I$</p> <p><u>Preservation:</u> $\{I \wedge b\} S \{I\}$</p> <p><u>Finalization:</u> $(I \wedge \neg b) \Rightarrow Q$</p>
--	---

27



Example 3

Use the invariant $I: Z=XJ$ to prove:

<pre> {true} Z := X J := 1 while J <> Y do Z := Z+X J := J+1 end_while {Z=XY} </pre>	<p><u>Initialization:</u> $P \Rightarrow I$</p> <p>What is "P" ?</p> <p style="padding-left: 40px;">$(Z=X \wedge J=1)$</p> <p>Does</p> <p>$(Z=X \wedge J=1) \Rightarrow Z=XJ$?</p> <p style="text-align: center;">Yes!</p>
--	--

28



Example 3

Use the invariant $I: Z=XJ$ to prove:

<pre> {true} Z := X J := 1 while J <> Y do Z := Z+X J := J+1 end_while {Z=XY} </pre>	<p>b</p> <p>\downarrow</p>	<p><u>Initialization:</u> $P \Rightarrow I$ ✓</p> <p><u>Preservation:</u> $\{I \wedge b\} S \{I\}$</p> <p>$\{Z=XJ \wedge J \neq Y\}$ $Z := Z+X$ $\{Z=X(J+1) \wedge J \neq Y\}$ $J := J+1$ $\{Z=X((J-1)+1) \wedge J-1 \neq Y\}$ $\Rightarrow Z=XJ$</p>
--	--	--

29



Example 3

Use the invariant $I: Z=XJ$ to prove:

<pre> {true} Z := X J := 1 while J <> Y do Z := Z+X J := J+1 end_while {Z=XY} </pre>	<p><u>Initialization:</u> $P \Rightarrow I$ ✓</p> <p><u>Preservation:</u> $\{I \wedge b\} S \{I\}$ ✓</p> <p><u>Finalization:</u> $(I \wedge \neg b) \Rightarrow Q$</p> <p>Does $(Z=XJ \wedge J=Y) \Rightarrow Z=XY$?</p> <p>Yes!</p>
--	--

30



Example 3

Use the invariant $I: Z=XJ$ to prove:

$\{true\}$	<u>Initialization</u> : $P \Rightarrow I$ ✓
$Z := X$	<u>Preservation</u> : $\{I \wedge b\} S \{I\}$ ✓
$J := 1$	<u>Finalization</u> : $(I \wedge \neg b) \Rightarrow Q$ ✓
while $J <> Y$ do	
$Z := Z+X$	
$J := J+1$	
end_while	
$\{Z=XY\}$	

31



Some Limitations of Formal Verification

- Difficulties can arise when dealing with:
 - parameters
 - pointers
 - synthesis of invariants
 - decidability of verification conditions
 - concurrency

32



Some Limitations of Formal Verification (cont.)

- In addition, a formal specification:
 - may be expensive to produce
 - may be incorrect and/or incomplete
 - normally reflects *functional* requirements only
- Will the proof process be manual or automatic? Who will prove the proof?