Introduction to Mathematical Logic

For CS Students

CS104/CS108

Yida TAO (陶伊达)

2024 年 5 月 28 日

南方科技大学

Table of Contents

Hoare Logic

1 Warm up

- To construct formal proofs of *partial correctness* specification, axioms and rules of inference are needed.
- This is what Hoare Logic provides:
  - The formulation of the deductive system is due to Hoare
  - Some of the underlying ideas originated with R. Floyd (also called Floyd–Hoare logic)
- A proof in Hoare logic is a sequence of lines, each of which is either an axiom of the logic or follows from earlier lines by a rule of inference of the logic
- A formal proof makes explicit what axioms and rules of inference are used to arrive at a conclusion.

Programs

We can think of any program of our core programming language as a sequence. All of the C_i below are either assignments, if-statements or while-statements. Of course, we allow the if-statements and while-statements to have embedded compositions.

$$
\begin{aligned}
&C_1;\\
&C_2;\\
&\quad .\\
&\quad .\\
&\quad .\\
&C_n
\end{aligned}
$$

Presentation of a proof

We should design a proof calculus which presents a proof of $\vdash_{par} (\!| \phi_0 |\!)\ P\ (\!| \phi_n |\!)$ by interleaving formulas with code as in

$$
\begin{array}{ll}
(\!| \phi_0 |\!) & \\
C_1; & \\
(\!| \phi_1 |\!) & \texttt{justification} \\
C_2; & \\
\cdot & \\
\cdot & \\
\cdot & \\
(\!| \phi_{n-1} |\!) & \texttt{justification} \\
C_n; & \\
(\!| \phi_n |\!) & \texttt{justification}
\end{array}
$$

Presentation of a proof

A full proof will have one or more conditions before and after each code statement. Each statement makes a Hoare triple with the preceding and following conditions. Each triple (postcondition) has a justification that explains its correctness.

```
{ program precondition }
y = 1;
{ ... }                          ⟨justification⟩
while (x != 0) {
    { ... }                      ⟨justification⟩
    y = y * x;
    { ... }                      ⟨justification⟩
    x = x - 1;
    { ... }                      ⟨justification⟩
}
{ program postcondition }        ⟨justification⟩
```

Table of Contents

2 Axioms and Inference Rules

Assignment

2 Axioms and Inference Rules

The rule for assignment has no premises and is therefore an axiom of our logic.

$$\overline{(\!| Q[E/x] |\!)\ x = E\ (\!| Q |\!)}$$

Intuition:
If we wish to show that Q holds in the state after $x = E$, we must show that Q holds before the assignment $x = E$, but with all free occurrences of x replaced by E in Q.

Assignment: Examples

2 Axioms and Inference Rules

What is the precondition ϕ?

- $(\!|\phi|\!)\ x = 2\ (\!|x = y|\!)$
- $(\!|\phi|\!)\ x = x + 1\ (\!|x = 2|\!)$
- $(\!|\phi|\!)\ x = y + z\ (\!|x = 1|\!)$
- $(\!|\phi|\!)\ x = x + 1\ (\!|x > 0 \wedge y > 0|\!)$

In program correctness proofs, we usually work backwards from the postcondition.

The implied rule for precondition strengthening:

$$\frac{P \to P' \qquad (\!| P' |\!)\ C\ (\!| Q |\!)}{(\!| P |\!)\ C\ (\!| Q |\!)}$$

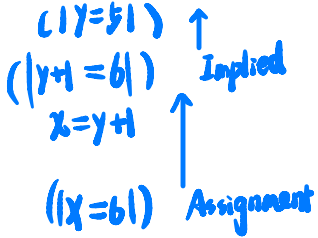
The implied rule for postcondition weakening:

$$\frac{(\!| P |\!)\ C\ (\!| Q' |\!) \qquad Q' \to Q}{(\!| P |\!)\ C\ (\!| Q |\!)}$$

The implied rule allows the precondition to be strengthened (i.e., we assume more than we need to), while the postcondition is weakened (i.e. we conclude less than we are entitled to).

Examples

2 Axioms and Inference Rules



Prove $\vdash_{par} \{\!|y = 5|\!\}\ x = y + 1\ \{\!|x = 6|\!\}$

$$\{\!|y = 5|\!\}$$
$$\{\!|y + 1 = 6|\!\} \qquad \text{Implied}$$
$$\texttt{x = y + 1}$$
$$\{\!|x = 6|\!\} \qquad \text{Assignment}$$

Although the proof is constructed bottom-up, its justifications make sense when read top-down.

Implied

2 Axioms and Inference Rules

"bridge"

The implied rule acts as a link between program logic and a suitable extension of FOL logic. It allows us to import proofs in predicate logic enlarged with the basic facts of arithmetic (e.g., $\forall x(x = x + 0)$, $r = x \wedge q = 0 \rightarrow r = x + y * q$), which are required for reasoning about integer expressions, into the proofs in program logic.

Composition

2 Axioms and Inference Rules

This rule is also known as the sequencing rule, which enables a partial correctness specification for a sequence $C_1; C_2$ to be derived from specification for C_1 and C_2.

$$\frac{(\!| P |\!)\ C_1\ (\!| Q |\!) \qquad (\!| Q |\!)\ C_2\ (\!| R |\!)}{(\!| P |\!)\ C_1; C_2\ (\!| R |\!)}$$

To prove $(\!| P |\!)\ C_1; C_2\ (\!| R |\!)$, we need to find appropriate midcondition Q and prove $(\!| P |\!)\ C_1\ (\!| Q |\!)$ and $(\!| Q |\!)\ C_2\ (\!| R |\!)$ (i.e., by splitting the problem into two.)

In our examples, the midcondition will usually be determined by a rule, such as the assignment rule.

Examples

2 Axioms and Inference Rules

1. $x = x_0 \wedge y = y_0$ premise
2. $y = y_0$ $\wedge e : 1$
3. $x = x_0$ $\wedge e : 1$
4. $y = y_0 \wedge x = x_0$ $\wedge i : 2, 3$

Prove $\vdash_{par} \{\!|\, x = x_0 \wedge y = y_0 \,|\!\}\ t = x; x = y; y = t\ \{\!|\, x = y_0 \wedge y = x_0 \,|\!\}$

$\{\!|\, ((x = x_0) \wedge (y = y_0)) \,|\!\}$

$\{\!|\, ((y = y_0) \wedge (x = x_0)) \,|\!\}$ implied *[proof required]*

```
t = x ;
```

$\{\!|\, ((y = y_0) \wedge (t = x_0)) \,|\!\}$ assignment

```
x = y ;
```

$\{\!|\, ((x = y_0) \wedge (t = x_0)) \,|\!\}$ assignment

```
y = t ;
```

$\{\!|\, ((x = y_0) \wedge (y = x_0)) \,|\!\}$ assignment

If statements

2 Axioms and Inference Rules

The proof rule for if-statements allows us to prove a triple of the form

$$(\!| P |\!)\ \text{if}\ B\ \{C_1\}\ \text{else}\ \{C_2\}\ (\!| Q |\!)$$

by decomposing it into two triples, subgoals corresponding to the cases of B evaluating to true and to false (i.e., the preconditions are augmented by the knowledge that B is true and false, respectively).

$$\frac{(\!| P \wedge B |\!)\ C_1\ (\!| Q |\!) \qquad (\!| P \wedge \neg B |\!)\ C_2\ (\!| Q |\!)}{(\!| P |\!)\ \text{if}\ B\ \{C_1\}\ \text{else}\ \{C_2\}\ (\!| Q |\!)}$$

If statements

2 Axioms and Inference Rules

分两种情况

$(\!\vert P \vert\!)$	
`if ( B ) {`	
$(\!\vert (P \wedge B) \vert\!)$	if-then-else
C_1	
$(\!\vert Q \vert\!)$	*(justify depending on C_1—a "subproof")*
`} else {`	
$(\!\vert (P \wedge (\neg B)) \vert\!)$	if-then-else
C_2	
$(\!\vert Q \vert\!)$	*(justify depending on C_2—a "subproof")*
`}`	
$(\!\vert Q \vert\!)$	if-then-else [*justifies this Q, given previous two*]

Examples

2 Axioms and Inference Rules

Prove the following is satisfied under partial correctness.

$\{\!|\ \texttt{true}\ |\!\}$

```
if ( x > y ) {
    max = x;
} else {
    max = y;
}
```

$\{\!|\ (((x > y) \wedge (max = x)) \vee ((x \leq y) \wedge (max = y)))\ |\!\}$

Examples

2 Axioms and Inference Rules

1. $x > y$ Premise
2. $x = x$
3. $x > y \wedge x = x$ $\wedge i : 1, 2$
4. $((x > y) \wedge (x = x)) \vee ((x \leq y) \wedge (x = y))$ $\vee i : 3$

```
{ true                                                          P
if ( x > y ) {                                                  B
    { (x > y) }                                                 P ∧ B
    { (((x > y) ∧ (x = x)) ∨ ((x ≤ y) ∧ (x = y))) }             implied (a)
    max = x ;
    { (((x > y) ∧ (max = x)) ∨ ((x ≤ y) ∧ (max = y))) }         assignment
} else {
    { (¬(x > y)) }                                              P ∧ ¬B
    { (((x > y) ∧ (y = x)) ∨ ((x ≤ y) ∧ (y = x))) }             implied (b)
    max = y ;
    { (((x > y) ∧ (max = x)) ∨ ((x ≤ y) ∧ (max = y))) }         assignment
}
{ (((x > y) ∧ (max = x)) ∨ ((x ≤ y) ∧ (max = y))) }             if-then-else
```

if-then-else

if-then-else

While statements

2 Axioms and Inference Rules

In the proof rule of partial-while (do not yet require termination):

- Loop invariant (循环不变式): I
- Premise: if I and B are true before we execute C, and C terminates, then I
- Conclusion: : no matter how many times the body C is executed, if I is true initially and the while statement terminates, then I will be true at the end. Moreover, since the while-statement has terminated, B will be false, $\neg B$ will be true.

$$\frac{(\!| I \wedge B |\!)\ C\ (\!| I |\!)}{(\!| I |\!)\ \text{while}\ B\ \{C\}\ (\!| I \wedge \neg B |\!)}$$

Loop Invariant

2 Axioms and Inference Rules

循环不变式

{I} while B {(I∧B)} C {(I)} ⇓ (I ∧ ¬B)

A loop invariant is:

- A relationship among the variables. (A predicate formula involving the variables.)
- The word “invariant” means something that does not change.
- It is true before the loop begins.
- It is true at the start of every iteration of the loop and at the end of every iteration of the loop.
- It is true after the loop ends.

$$\frac{(\!| I \wedge B |\!)\ C\ (\!| I |\!)}{(\!| I |\!)\ \text{while}\ (B)\ \{C\}\ (\!| I \wedge \neg B |\!)} \quad \text{(partial while)}$$

Proving partial correctness of a while loop

Steps to follow:

- Find a loop invariant.
- Complete the annotations.
- Prove any implied's.

Example I

For the following program C:

```
z=1;
while (z*z<16) {
    z=z+1;
}
```

Find a loop invariant to prove $\vdash_{par} (\!| true |\!)\ C\ (\!| z = 4 |\!)$.

Example I

2 Axioms and Inference Rules

- Loop invariant candidate 1: $z \geq 1$
- This loop invariant $z \geq 1$ is not useful, cannot prove Implied (b).

$$\{\!|\ \text{true}\ |\!\}$$
`z = 1;`
$$\{\!|\ (z \geq 1)\ |\!\}$$ — Assignment

`while (z * z < 16)`
$$\{\!|\ ((z \geq 1) \wedge ((z \cdot z) < 16))\ |\!\}$$ — Partial-While

`z = z + 1;`
$$\{\!|\ (z \geq 1)\ |\!\}$$
`}`
$$\{\!|\ ((z \geq 1) \wedge (\neg((z \cdot z) < 16)))\ |\!\}$$ — Partial-While
$$\{\!|\ (z = 4)\ |\!\}$$ — ???

Example I

2 Axioms and Inference Rules

- Loop invariant candidate 2: $z * z \leq 16$
- This loop invariant $z * z \leq 16$ is not useful, cannot prove Implied (b), since z might be -4.

$\{\!|\ \text{true}\ |\!\}$
`z = 1;`
$\{\!|\ ((z \cdot z) \leq 16)\ |\!\}$ Assignment
`while (z * z < 16){`
  $\{\!|\ (((z \cdot z) \leq 16) \wedge ((z \cdot z) < 16))\ |\!\}$ Partial-While

  `z = z + 1;`
  $\{\!|\ ((z \cdot z) \leq 16)\ |\!\}$
`}`
$\{\!|\ (((z \cdot z) \leq 16) \wedge (\neg((z \cdot z) < 16)))\ |\!\}$ Partial-While
$\{\!|\ (z = 4)\ |\!\}$???

Example I

2 Axioms and Inference Rules

Combine both invariants: $(z \geq 1) \wedge (z * z \leq 16)$

$\{\!|\ \text{true}\ |\!\}$

$\{\!|\ ((1 \geq 1) \wedge ((1 \cdot 1) \leq 16))\ |\!\}$ Implied(a)

```
z = 1;
```

$\{\!|\ ((z \geq 1) \wedge ((z \cdot z) \leq 16))\ |\!\}$ Assignment

```
while (z * z < 16){
```

$\{\!|\ (((z \geq 1) \wedge ((z \cdot z) \leq 16)) \wedge ((z \cdot z) < 16))\ |\!\}$ Partial-While

$\{\!|\ (((z+1) \geq 1) \wedge (((z+1) \cdot (z+1)) \leq 16))\ |\!\}$ Implied (b)

```
    z = z + 1;
```

$\{\!|\ ((z \geq 1) \wedge ((z \cdot z) \leq 16))\ |\!\}$ Assignment

```
}
```

$\{\!|\ (((z \geq 1) \wedge ((z \cdot z) \leq 16)) \wedge (\neg((z \cdot z) < 16)))\ |\!\}$ Partial-While

$\{\!|\ (z = 4)\ |\!\}$ Implied (c)

Example II

2 Axioms and Inference Rules

Prove that the following triple is satisfied under partial correctness.

$(\!| (x \geq 0) |\!)$

```
y = 1 ;
z = 0 ;
while (z != x) {
    z = z + 1 ;
    y = y * z ;
}
```

$(\!| (y = x!) |\!)$

Example II

Step 1: Write down the values of all the variables every time the `while` test is reached.

$\{\!|\ (x \geq 0)\ |\!\}$

```
y = 1 ;
z = 0 ;
while (z != x) {
    z = z + 1 ;
    y = y * z ;
}
```

$\{\!|\ (y = x!)\ |\!\}$

At the `while` statement:

x	y	z	$z \neq x$
5	1	0	true
5	1	1	true
5	2	2	true
5	6	3	true
5	24	4	true
5	120	5	false

Example II

Step 2: Find relationships among the variables that are true for every `while` test. These are our candidate invariants.

$\{\!|\ (x \geq 0)\ |\!\}$

```
y = 1 ;
z = 0 ;
while (z != x) {
    z = z + 1 ;
    y = y * z ;
}
```

$\{\!|\ (y = x!)\ |\!\}$

At the `while` statement:

x	y	z	$z \neq x$
5	1	0	true
5	1	1	true
5	2	2	true
5	6	3	true
5	24	4	true
5	120	5	false

Example II

Is $\neg(z = x)$ a loop invariant? ✗ be true all the time

$$\{\!|\ (x \geq 0)\ |\!\}$$

```
y = 1 ;
z = 0 ;
while (z != x) {
    z = z + 1 ;
    y = y * z ;
}
```

$$\{\!|\ (y = x!)\ |\!\}$$

At the `while` statement:

x	y	z	$z \neq x$
5	1	0	true
5	1	1	true
5	2	2	true
5	6	3	true
5	24	4	true
5	120	5	false

Example II

Is $x \geq 0$ a loop invariant? X 与y,z无联系

$\{\!|\ (x \geq 0)\ |\!\}$

```
y = 1 ;
z = 0 ;
while (z != x) {
    z = z + 1 ;
    y = y * z ;
}
```

$\{\!|\ (y = x!)\ |\!\}$

At the `while` statement:

x	y	z	$z \neq x$
5	1	0	true
5	1	1	true
5	2	2	true
5	6	3	true
5	24	4	true
5	120	5	false

Example II

Is $y \geq z$ a loop invariant?

not useful

$$
\begin{array}{l}
\{\!|\ (x \geq 0)\ |\!\} \\
\texttt{y = 1 ;} \\
\texttt{z = 0 ;} \\
\texttt{while (z != x) \{} \\
\quad \texttt{z = z + 1 ;} \\
\quad \texttt{y = y * z ;} \\
\texttt{\}} \\
\{\!|\ (y = x!)\ |\!\}
\end{array}
$$

At the `while` statement:

x	y	z	$z \neq x$
5	1	0	true
5	1	1	true
5	2	2	true
5	6	3	true
5	24	4	true
5	120	5	false

Example II

Is $y = z!$ a loop invariant?

$(\!| (x \geq 0) |\!)$

```
y = 1 ;
z = 0 ;
while (z != x) {
    z = z + 1 ;
    y = y * z ;
}
```

$(\!| (y = x!) |\!)$

At the `while` statement:

x	y	z	$z \neq x$
5	1	0	true
5	1	1	true
5	2	2	true
5	6	3	true
5	24	4	true
5	120	5	false

Example II

Step 3: Try each candidate invariant until we find one that works for our proof.

$\{\!|\ (x \geq 0)\ |\!\}$

```
y = 1 ;
z = 0 ;
while (z != x) {
    z = z + 1 ;
    y = y * z ;
}
```

$\{\!|\ (y = x!)\ |\!\}$

At the `while` statement:

x	y	z	$z \neq x$
5	1	0	true
5	1	1	true
5	2	2	true
5	6	3	true
5	24	4	true
5	120	5	false

Example II

2 Axioms and Inference Rules

First, annotate by partial-while, with the chose loop invariant $y = z!$.

我到 恒成立的 有效关系 I

```
(| x ≥ 0 |)

y = 1 ;

z = 0 ;
(| y = z! |)                                  [justification required]
while (z != x) {
    (| (y = z!) ∧ ¬(z = x) |)                 partial-while ((| I ∧ B |))

    z = z + 1 ;

    y = y * z ;
    (| y = z! |)                              [justification required]
}                          I ∧ ¬B
(| y = z! ∧ (z = x) |)                        partial-while ((| I ∧ ¬B |))
(| y = x! |)
```

Example II

2 Axioms and Inference Rules

Next, annotate assignment statements.

$\{\!|\, x \geq 0 \,|\!\}$
$\{\!|\, 1 = 0! \,|\!\}$
`y = 1 ;`
$\{\!|\, y = 0! \,|\!\}$ assignment
`z = 0 ;`
$\{\!|\, y = z! \,|\!\}$ assignment
`while (z != x) {`
  $\{\!|\, (y = z!) \wedge \neg(z = x) \,|\!\}$ partial-while
  $\{\!|\, y(z+1) = (z+1)! \,|\!\}$
  `z = z + 1 ;`
  $\{\!|\, yz = z! \,|\!\}$ assignment
  `y = y * z ;`
  $\{\!|\, y = z! \,|\!\}$ assignment
`}`
$\{\!|\, y = z! \wedge (z = x) \,|\!\}$ partial-while
$\{\!|\, y = x! \,|\!\}$

Example II

2 Axioms and Inference Rules

Then, note the implied, to be proven separately.

$(\!| x \geq 0 |\!)$

$(\!| 1 = 0! |\!)$ implied (a)

```
y = 1 ;
```

$(\!| y = 0! |\!)$ assignment

```
z = 0 ;
```

$(\!| y = z! |\!)$ assignment

```
while (z != x) {
```

$(\!| (y = z!) \wedge \neg(z = x) |\!)$ partial-while

$(\!| y(z+1) = (z+1)! |\!)$ implied (b)

```
    z = z + 1 ;
```

$(\!| yz = z! |\!)$ assignment

```
    y = y * z ;
```

$(\!| y = z! |\!)$ assignment

```
}
```

$(\!| y = z! \wedge (z = x) |\!)$ partial-while

$(\!| y = x! |\!)$ implied (c)

Example II

2 Axioms and Inference Rules

Finally, prove the implied assertions using the inference rules of ordinary logic (FOL, arithmetic).

Proof of implied (a): $(x \geq 0) \vdash (1 = 0!)$
By definition of factorial.

$\{\!|\ x \geq 0\ |\!\}$
$\{\!|\ 1 = 0!\ |\!\}$ implied (a)
`y = 1 ;`
$\{\!|\ y = 0!\ |\!\}$ assignment
`z = 0 ;`
$\{\!|\ y = z!\ |\!\}$ assignment
`while (z != x) {`
  $\{\!|\ (y = z!) \wedge \neg(z = x)\ |\!\}$ partial-while
  $\{\!|\ y(z+1) = (z+1)!\ |\!\}$ implied (b)
  `z = z + 1 ;`
  $\{\!|\ yz = z!\ |\!\}$ assignment
  `y = y * z ;`
  $\{\!|\ y = z!\ |\!\}$ assignment
`}`
$\{\!|\ y = z! \wedge (z = x)\ |\!\}$ partial-while
$\{\!|\ y = x!\ |\!\}$ implied (c)

Example II

2 Axioms and Inference Rules

Proof of implied (c):

$$(y = z!) \wedge (z = x) \vdash (y = x!)$$

1.	$(y = z!) \wedge (z = x)$	Premise
2.	$(y = z!)$	$\wedge$e:1
3.	$(z = x)$	$\wedge$e:1
4.	$(z! = x!)$	eq. of substitution 3
5.	$(y = x!)$	transitivity of eq. 2,4

$\{\!| x \geq 0 |\!\}$

$\{\!| 1 = 0! |\!\}$ implied (a)

```
y = 1 ;
```

$\{\!| y = 0! |\!\}$ assignment

```
z = 0 ;
```

$\{\!| y = z! |\!\}$ assignment

```
while (z != x) {
```

$\{\!| (y = z!) \wedge \neg(z = x) |\!\}$ partial-while

$\{\!| y(z+1) = (z+1)! |\!\}$ implied (b)

```
    z = z + 1 ;
```

$\{\!| yz = z! |\!\}$ assignment

```
    y = y * z ;
```

$\{\!| y = z! |\!\}$ assignment

```
}
```

$\{\!| y = z! \wedge (z = x) |\!\}$ partial-while

$\{\!| y = x! |\!\}$ implied (c)

Example II

2 Axioms and Inference Rules

Proof of implied (b):

$$(y = z!) \wedge \neg(z = x) \vdash (z + 1)y = (z + 1)!$$

1.	$(y = z!) \wedge \neg(z = x)$	Premise
2.	$(y = z!)$	$\wedge$e:1
3.	$(z + 1)y = (z + 1)z!$	eq. of subs 2
4.	$(z + 1)z! = (z + 1)!$	def. of factorial 3
5.	$(z + 1)y = (z + 1)!$	trans of eq. 3,4

$\{\!| x \geq 0 |\!\}$

$\{\!| 1 = 0! |\!\}$ implied (a)

```
y = 1 ;
```

$\{\!| y = 0! |\!\}$ assignment

```
z = 0 ;
```

$\{\!| y = z! |\!\}$ assignment

```
while (z != x) {
```

$\{\!| (y = z!) \wedge \neg(z = x) |\!\}$ partial-while

$\{\!| y(z + 1) = (z + 1)! |\!\}$ implied (b)

```
    z = z + 1 ;
```

$\{\!| yz = z! |\!\}$ assignment

```
    y = y * z ;
```

$\{\!| y = z! |\!\}$ assignment

```
}
```

$\{\!| y = z! \wedge (z = x) |\!\}$ partial-while

$\{\!| y = x! |\!\}$ implied (c)

Example III

Prove the following is satisfied under partial correctness.

$$\{\!|\ ((n \geq 0) \wedge (a \geq 0))\ |\!\}$$

```
s = 1 ;
i = 0 ;
while (i < n) {
    s = s * a ;
    i = i + 1 ;
}
```

$$\{\!|\ (s = a^n)\ |\!\}$$

Example III

Step 1: Draw an execution trace to help find the invariant.

$(\!| ((n \geq 0) \wedge (a \geq 0)) |\!)$

```
s = 1 ;
i = 0 ;
while (i < n) {
    s = s * a ;
    i = i + 1 ;
}
```

$(\!| (s = a^n) |\!)$

Trace of the loop:

a	n	i	s
2	3	0	1
2	3	1	$1*2$
2	3	2	$1*2*2$
2	3	3	$1*2*2*2$

Example III

2 Axioms and Inference Rules

Attempt 1: try the invariant $s = a^i$.

But implied (c) cannot be proved.

We must use a different invariant.

$\{\!|\ ((n \geq 0) \wedge (a \geq 0))\ |\!\}$
$\{\!|\ \ldots\ |\!\}$
`s = 1 ;`
$\{\!|\ \ldots\ |\!\}$
`i = 0 ;`
$\{\!|\ (s = a^i)\ |\!\}$
`while (i < n) {`
  $\{\!|\ ((s = a^i) \wedge (i < n))\ |\!\}$ partial-while
  $\{\!|\ \ldots\ |\!\}$
  `s = s * a ;`
  $\{\!|\ \ldots\ |\!\}$
  `i = i + 1 ;`
  $\{\!|\ (s = a^i)\ |\!\}$
`}`
$\{\!|\ ((s = a^i) \wedge (i \geq n))\ |\!\}$ partial-while
$\{\!|\ (s = a^n)\ |\!\}$ implied (c)

Example III

2 Axioms and Inference Rules

Attempt 2: try the invariant $(s = a^i) \wedge (i \leq n)$.

Now, the proof succeeds.

$\{\!|\ ((n \geq 0) \wedge (a \geq 0))\ |\!\}$

$\{\!|\ ((1 = a^0) \wedge (0 \leq n))\ |\!\}$ implied (a)

```
s = 1 ;
```

$\{\!|\ ((s = a^0) \wedge (0 \leq n))\ |\!\}$ assignment

```
i = 0 ;
```

$\{\!|\ \big((s = a^i) \wedge (i \leq n)\big)\ |\!\}$ assignment

```
while (i < n) {
```

$\{\!|\ \big(\big((s = a^i) \wedge (i \leq n)\big) \wedge (i < n)\big)\ |\!\}$ partial-while

$\{\!|\ (((s \cdot a) = a^{i+1}) \wedge ((i + 1) \leq n))\ |\!\}$ implied (b)

```
    s = s * a ;
```

$\{\!|\ ((s = a^{i+1}) \wedge ((i + 1) \leq n))\ |\!\}$ assignment

```
    i = i + 1 ;
```

$\{\!|\ \big((s = a^i) \wedge (i \leq n)\big)\ |\!\}$ assignment

```
}
```

$\{\!|\ \big(\big((s = a^i) \wedge (i \leq n)\big) \wedge (i \geq n)\big)\ |\!\}$ partial-while

$\{\!|\ (s = a^n)\ |\!\}$ implied (c)

A note on loop invariant

The discovery of a suitable invariant:

- a necessary step in order to use the proof rule Partial-while.
- in general it requires intelligence and ingenuity
- This contrasts markedly with the case of the proof rules for if-statements and assignments, which are purely mechanical in nature: their usage is just a matter of symbol-pushing and does not require any deeper insight.

Table of Contents

3 Proof tableaux

Definition

The process of obtaining ϕ_i from C_{i+1} and ϕ_{i+1} is called computing the weakest precondition of C_{i+1}, given the postcondition ϕ_{i+1}.

That is to say, we are looking for the logically weakest formula whose truth at the beginning of the execution of C_{i+1} is enough to guarantee ϕ_{i+1}.

General Construction Process

3 Proof tableaux

The construction of a proof tableau for $(\!|\, \phi_i \,|\!)\ C_1; \ldots; C_n\ (\!|\, \psi \,|\!)$:

- Starting with the postcondition ψ
- pushing it upwards through C_n, then C_{n-1}, ... , until a formula ϕ' emerges at the top.
- ϕ' is then checked to see whether it follows from the given precondition ϕ, applying the *Implied* rule.

Examples (Assignment and Implied)

Proof the validity.

- $\vdash_{par} (\!| y = 5 |\!)\ x = y + 1\ (\!| x = 6 |\!)$
- $\vdash_{par} (\!| y < 3 |\!)\ y = y + 1\ (\!| y < 4 |\!)$
- $\vdash_{par} (\!| T |\!)\ z = x; z = z + y; u = z\ (\!| u = x + y |\!)$

Although the proof is constructed bottom-up, its justifications make sense when read top-down.

Construction Process for If Statement

Suppose we are given a condition ψ and a program fragment if B $\{C_1\}$ else $\{C_2\}$. We want to calculate the weakest ϕ such that:

$$(\!|\, \phi \,|\!) \text{ if } B \ \{C_1\} \text{ else } \{C_2\} \ (\!|\, \psi \,|\!)$$

ϕ may be calculated as follows:

- Push ψ upwards through C_1; call the result ϕ_1.
- Push ψ upwards through C_2; call the result ϕ_2.
- Set ϕ to be $(B \to \phi_1) \wedge (\neg B \to \phi_2)$

Example

For the below code P

```
a=x+1;
if(a-1==0){
    y=1;
} else {
    y=a;
}
```

Show that $\vdash_{par} (\!| T |\!)\ P\ (\!| y = x + 1 |\!)$ is valid.

Readings

3 Proof tableaux

- Text B: chapter 4.3.2
- Reference: lecture notes of CS245, University of Waterloo.

Introduction to Mathematical Logic

Thank you for listening!
Any questions?