Proving Program Correctness

The Axiomatic Approach

What is Correctness?

- Correctness:
  - partial correctness + termination
- Partial correctness:
  - Program implements its specification

Proving Partial Correctness

- Goal: prove that program is partially correct
- Approach: model computation with predicates
  - Predicates are boolean functions over program state
- Simple example
  - {odd(x)} a = x {odd(a)}
- Generally: {P} S {Q}, where
  - P is a precondition
  - Q is a postcondition
  - S is a set of programming language statements

Proof System

- Two elements of proof system
- Axioms
  - Capture the effect of individual programming language statements
- Inference rules
  - Compose the effect of individual statements and extrinsic knowledge to build up proofs of entire program

Axioms

- Axioms explain the effect of executing a single statement
  - **Assignment**
  - If
  - If then else
  - While loop

Assignment Axiom

- Rule: $\{P^x_y\}\ x \leftarrow y\ \{P\}$
- Application: Replace all free occurences of x with y
  - e.g., {odd(x)} a = x {odd(a)}

Inference Rules

- Inference rules allow us to compose the effects of individual statements and extrinsic knowledge to build up proofs of entire program
- 3 inference rules
  - Composition
  - Consequence 1
  - Consequence 2

Composition

- Rule: $$\frac{\{P\}\, S_1 \{Q\}\,,\, \{Q\}\, S_2 \{R\}}{\{P\}\, S_1 \,;\, S_2 \,\{R\}}$$
- Consider two predicates
  - {odd(x)} a = x {odd(a)}
  - {odd(x+1)} x = x+1 {odd(x)}
- What is the effect of executing both stmts?
  - {odd(x+1)} x = x+1 ; a = x {odd(a)}

Consequence 1

- Rule $$\frac{\{P\}\ S\ \{R\}\ ,\ R \Rightarrow Q}{\{P\}\ S\ \{Q\}}$$
- Ex:
  - {odd(x)} a = x {odd(a)} and
  - Postcondition Q ≡ {a ≠ 4}
- What can we say about this program?

$$\frac{\{\text{odd}(x)\}\ a = x\ \{\text{odd}(a)\}\ ,\ \ \text{odd}\ (a) \Rightarrow a \neq 4}{\{\text{odd}(x)\}\ a = x\ \{a \neq 4\}}$$

Consequence 2

- Rule: $$\frac{P \Rightarrow R\ ,\ \{R\}\ S\ \{Q\}}{\{P\}\ S\ \{Q\}}$$
- Ex:
  - Precondition P ≡ {x=1} and
  - {odd(x)} a = x {odd(a)}
- What can we say about this program?

$$\frac{x = 1 \Rightarrow \text{odd}(x)\ ,\ \{\text{odd}(x)\}\ a = x\ \{\text{odd}(a)\}}{\{x = 1\}\ a = x\ \{\text{odd}(a)\}}$$

Axioms (cont.)

- Axioms explain the effect of executing a single statement
  - Assignment
  - **If**
  - **If then else**
  - **While loop**

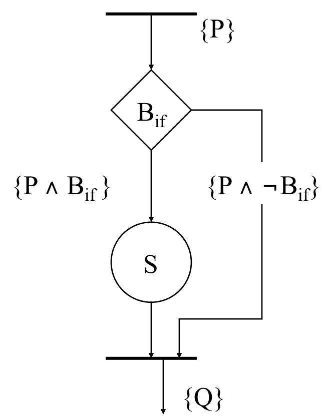
Assignment Axiom

- Rule: $\{P^x_y\}\ x \leftarrow y\ \{P\}$
- Application: Replace all free occurences of x with y
  - e.g., {odd(x)} a = x {odd(a)}

if Axiom

- Rule:

$$\frac{\{P \wedge B_{if}\}\, S\, \{Q\},\ \{P \wedge \neg B_{if}\} \Rightarrow \{Q\}}{\{P\}\ \text{if } B_{if} \text{ then } S\ \{Q\}}$$



Application

- Example:
  1. if even(x) then {
  2. x = x +1
  3. }

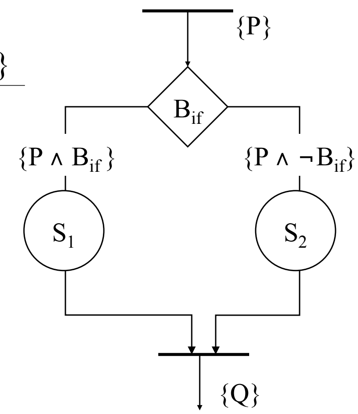
  {odd(x) ∧ x > 3}

- else part: need to show
  - $\{(P \wedge \neg\text{even}(x)) \Rightarrow (\text{odd}(x) \wedge x>3)\}$
  - $\{P \Rightarrow (x>3)\}$
- then part: need to show
  - $\{P \wedge \text{even}(x)\}\ x=x+1\ \{\text{odd}(x) \wedge x>3\}$
  - $\{\text{odd}(x+1) \wedge x>2\}\ x = x+1\ \{\text{odd}(x) \wedge x > 3\}$
  - $\{(P \wedge \text{even}(x)) \Rightarrow (\text{odd}(x+1) \wedge x>2)\}$
  - $\{P \Rightarrow (x>2)\}$
- Need to choose a predicate P consistent with implications above
- $P \equiv x>2$
  - $x > 39$ works as well

if then else Axiom

- Rule

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



Conditional Stmt 2 Axiom

- Example:
  1. if x < 0 then {
  2. x = -x;
  3. y = x
  4. } else {
  5. y = x
  6. }

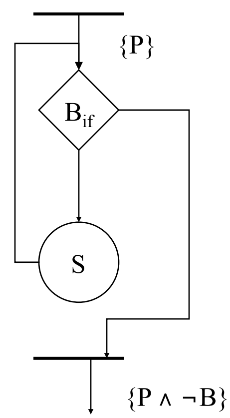
  {y = |x|}

- Then part: need to show
  - $\{P \wedge (x<0)\}$ x=-x;y=x $\{y = |x|\}$
  - $\{x = |x|\}$ y = x $\{y = |x|\}$
  - $\{-x = |x|\}$ x = -x $\{x = |x|\}$
  - $(P \wedge x < 0) \Rightarrow -x = |x|$
- Else part: need to show
  - $\{P \wedge \neg (x<0)\}$ y=x $\{y = |x|\}$
  - $\{x = |x|\}$ y=x $\{y=|x|\}$
  - $(P \wedge \neg(x < 0)) \Rightarrow x = |x|$
- $P \equiv$ true

While Loop Axiom

- Rule

$$\frac{\{P \wedge B\}\ S\ \{P\}}{\{P\}\ \text{while } B \text{ do } S\ \{P \wedge \neg B\}}$$



- Infinite number of paths, so we need one predicate for that captures the effect of 0 or more loop traversals
- P is called a loop invariant

Proving Partial Correctness

- Handle termination separately
- Axioms and inference rules are applied in reverse during proof
  - Start with postcondition and work backwards to determine what must precondition must be

Partial Correctness Proof

IN ≡ $\{B \geq 0\}$

```
a = A
b = B
y = 0
while b > 0 do {
    y = y + a
    b = b - 1
}
```

OUT ≡ $\{y = AB\}$

While Loop

IN ≡ $\{B \geq 0\}$

```
a = A
b = B
y = 0
while b > 0 do {
   y = y + a
   b = b - 1
}
```

OUT ≡ $\{y = AB\}$

- From while loop axiom need to show $\{P \wedge B\}$ S $\{P\}$
- $P \equiv y + ab = AB \wedge b \geq 0$
- $B_w \equiv b > 0$
- $\{y + ab = AB \wedge b \geq 0\}$ y=y+a; b=b-1 $\{P\}$
- $\{y+a(b-1) = AB \wedge b-1 \geq 0\}$ $b = b - 1$ $\{P\}$
- $\{y+a+a(b-1) = AB \wedge b-1 \geq 0\}$ $y = y+a$ $\{\ldots.\}$
- $\{y + ab = AB \wedge b-1 \geq 0\}$ loop body $\{P\}$
- $\{y + ab = AB \wedge b \geq 0 \wedge b > 0\}$
- $\Rightarrow \{y + ab = AB \wedge b-1 \geq 0\}$
- From while loop axiom can conclude $\{P\}$ while loop $\{P \wedge \neg B_w\}$

While Loop

$IN \equiv \{B \geq 0\}$

```
a = A
b = B
y = 0
while b > 0 do {
   y = y + a
   b = b - 1
}
```

$OUT \equiv \{y = AB\}$

- Now need to show $P \wedge \neg B_w \Rightarrow OUT$
- $P \equiv y + ab = AB \wedge b \geq 0$
- $Bw \equiv b > 0$

- $y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)$
- $y + ab = AB \wedge b = 0$
- $y = AB$
- So $\{P \wedge \neg Bw\} \Rightarrow OUT$

- From consequence rule we can conclude {P} while loop {OUT}

While Loop

$IN \equiv \{B \geq 0\}$

```
a = A
b = B
y = 0
while b > 0 do {
   y = y + a
   b = b - 1
}
```

$OUT \equiv \{y = AB\}$

- $P \equiv y + ab = AB \wedge b \geq 0$
- Establish {IN} a=A;b=B;y=0 {P}

- $\{ab = AB \wedge b \geq 0\}$ y=0 { P}
- $\{aB = AB \wedge B \geq 0\}$ b = B {….}
- $\{AB = AB \wedge B \geq 0\}$ a = A {….}
- So {IN} a=A;b=B;y=0 {P}

While Loop Axiom

- So
  - $\{IN\}$ lines 1-3 $\{P\}$,
  - $\{P\}$ while loop $\{P \wedge \neg B_w\}$, and
  - $\{P \wedge \neg B_w\} \Rightarrow OUT$
- Therefore
  - $\{IN\}$ program $\{OUT\}$

Total correctness

- After you have shown partial correctness
  - Need to prove that program terminates
- Usually a progress argument. For previous program
  - Loop terminates if $b \leq 0$
  - b starts positive and is decremented by 1 every iteration
  - So loop must eventually terminate

Now You Try It

```
r = 1;
i = 0;
while i < m do {
    r = r  * n;
    i = i + 1
}
```

Postcondition: $r = n^m$