CS 599: Formal Methods in Security and Privacy
Hoare Triples and Hoare Logic

Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu
Programming Language

c ::= abort
   | skip
   | x := e
   | c ; c
   | if e then c else c
   | while e do c

\[
x, y, z, \ldots \quad \text{program variables}
\]
\[
e_1, e_2, \ldots \quad \text{expressions}
\]
\[
c_1, c_2, \ldots \quad \text{commands}
\]
Specifications - Hoare triple

Precondition
Program
Postcondition

\[ c : P \implies Q \]

Precondition (a logical formula)

Program

Postcondition (a logical formula)
Some examples

Precondition

\[ x := z + 1 : \{ z + 1 > 0 \} \Rightarrow \{ x > 0 \} \]

Postcondition

Is it a good specification?
Some examples

\[ x := z + 1 : \{ z + 1 > 0 \} \Rightarrow \{ x > 0 \} \]

Precondition

Is it a good specification?

Postcondition

✓
Some examples

\begin{align*}
\text{i} & := 0; \\
\text{r} & := 1; \\
\text{while} (i \leq k) \text{do} & \\
\quad \text{r} & := \text{r} \times \text{n}; \\
\quad \text{i} & := \text{i} + 1
\end{align*}

\begin{align*}
\text{Precondition} & : \{ 0 \leq k \} \Rightarrow \{ r = n^k \} \\
\text{Postcondition} & \end{align*}

Is it a good specification?
Some examples

\[
\begin{align*}
i &:= 0; \\
r &:= 1; \\
\text{while } (i \leq k) \text{ do} & \quad r := r \times n; \\
& \quad i := i + 1
\end{align*}
\]

Precondition

\[
\{ 0 \leq k \} \Rightarrow \{ r = n^k \}
\]

Postcondition

Is it a good specification?

✗
Some examples

- Precondition:
  \[ \{ 0 \leq k \} \Rightarrow \{ r = n^k \} \]

- Postcondition:

- Is it a good specification?

- Initial state:
  \( m_{in} = [k = 0, n = 2, i = 0, r = 0] \)

- Final state:
  \( m_{out} = [k = 0, n = 2, i = 1, r = 2] \)

- Code snippet:

  ```
  i := 0;
  r := 1;
  while (i <= k) do
      r := r * n;
      i := i + 1
  ```
Some examples

\[
i := 0; \\
r := 1; \\
\text{while}(i \leq k) \text{do} \\
\quad r := r \times n; \\
i := i + 1
\]

Precondition

\[
\{ 0 < k \} \Rightarrow \{ r = n^k \}
\]

Postcondition

Is it a good specification?
Some examples

Precondition:

\[ \{0 < k\} \Rightarrow \{r = n^k\} \]

Postcondition:

Is it a good specification? X

i := 0;
r := 1;
while (i ≤ k) do
  r := r * n;
i := i + 1
Some examples

\[i := 0;\]
\[r := 1;\]
\[\text{while}(i \leq k) \text{do}\]
\[r := r \times n;\]
\[i := i + 1\]

Precondition
\[\{0 < k\} \Rightarrow \{r = n^k\}\]

Postcondition

Is it a good specification?

\[m_{in} = [k = 1, n = 2, i = 0, r = 0]\]
\[m_{out} = [k = 1, n = 2, i = 2, r = 4]\]
Some examples

i := 0;
r := 1;
while (i < k) do
  r := r * n;
i := i + 1

Precondition

\[ \{ 0 \leq k \} \Rightarrow \{ r = n^k \} \]

Postcondition

Is it a good specification?
Some examples

\[ \{0 \leq k\} \Rightarrow \{r = n^k\} \]

\[
\begin{align*}
i &: = 0; \\
r &:= 1; \\
\text{while}(i < k)\text{do} \\
&\quad r := r \times n; \\
&\quad i := i + 1
\end{align*}
\]

Is it a good specification? ✅
Some examples

```
i := 0;
 r := 1;
while (i ≤ k) do
  r := r * n;
i := i + 1
```

Precondition

\[
\{ 0 \leq k \} \Rightarrow \{ r = n^i \}
\]

Is it a good specification?
Some examples

Precondition:
\[
\{0 \leq k\} \Rightarrow \{r = n^i\}
\]

Postcondition:

Is it a good specification?

```
i:=0;
r:=1;
while(i\leq k) do
    r:=r \times n;
i:=i + 1
```
Some examples

\[
\begin{align*}
i &:= 0; \\
r &:= 1; \\
\text{while}(i \leq k)\text{do} & \\
\quad r &:= r \ast n; \\
i &:= i + 1
\end{align*}
\]

Precondition

\[\{0 < k \land k < 0\} \Rightarrow \{r = n^k\}\]

Postcondition

Is it a good specification?
Some examples

```
i := 0;
r := 1;
while (i <= k) do
    r := r * n;
    i := i + 1
```

Precondition

\[ \{ 0 < k \land k < 0 \} \Rightarrow \{ r = n^k \} \]

Postcondition

Is it a good specification? ✓
Some examples

\[
\begin{align*}
i &:= 0; \\
r &:= 1; \\
\text{while} (i \leq k) \text{do} \\
& \quad r := r \times n; \\
& \quad i := i + 1
\end{align*}
\]

Precondition

\[
\{ 0 < k \wedge k < 0 \} \Rightarrow \{ r = n^k \}
\]

Postcondition

Is it a good specification?

\[
\begin{array}{c}
\checkmark
\end{array}
\]

This is good because there is no memory that satisfies the precondition.
How do we determine the validity of an Hoare triple?
Validity of Hoare triple

\[ c : P \implies Q \]

- **Precondition** (a logical formula)
- **Program**
- **Postcondition** (a logical formula)
Validity of Hoare triple

We are interested only in inputs that meets $P$ and we want to have outputs satisfying $Q$. 

$c : P \Rightarrow Q$ 

- **Precondition** (a logical formula) 
- **Postcondition** (a logical formula) 
- **Program**
Validity of Hoare triple

We are interested only in inputs that meets P and we want to have outputs satisfying Q.

How shall we formalize this intuition?
Validity of Hoare triple
We say that the triple $c: P \Rightarrow Q$ is valid if and only if for every memory $m$ such that $P(m)$ and memory $m'$ such that $\{c\}_m = m'$ we have $Q(m')$. 
Validity of Hoare triple

We say that the triple $c: P \Rightarrow Q$ is valid if and only if for every memory $m$ such that $P(m)$ and memory $m'$ such that $\{c\}_m = m'$ we have $Q(m')$.

Is this condition easy to check?
Hoare Logic
Floyd-Hoare reasoning

A verification of an interpretation of a flowchart is a proof that for every command $c$ of the flowchart, if control should enter the command by an entrance $a_i$ with $P_i$ true, then control must leave the command, if at all, by an exit $b_j$ with $Q_j$ true. A semantic definition of a particular set of command types, then, is a rule for constructing, for any command $c$ of one of these types, a verification condition $V_c(P; Q)$ on the antecedents and consequents of $c$. This verification condition must be so constructed that a proof that the verification condition is satisfied for the antecedents and consequents of each command in a flowchart is a verification of the interpreted flowchart.
Rules of Hoare Logic: Skip

⊢ skip: P ⇒ P
Rules of Hoare Logic: Skip

\[ \vdash \text{skip} : \ P \Rightarrow P \]

Is this correct?
We say that an axiom is correct if we can prove the validity of each triple which is an instance of the conclusion.
Correctness of Skip Rule

\[ \vdash \text{skip: } P \Rightarrow P \]

To show this rule correct we need to show the validity of the triple skip: \( P \Rightarrow P \).
Correctness of Skip Rule

⊢ skip: P ⇒ P

To show this rule correct we need to show the validity of the triple skip: P ⇒ P.

For every m such that P(m) and m’ such that \{skip\}_m=m’ we need P(m’).
Correctness of Skip Rule

⊢ skip: P ⇒ P

To show this rule correct we need to show the validity of the triple skip: P ⇒ P.

For every m such that P(m) and m’ such that \{skip\}_m = m’ we need P(m’).

Follow easily by our semantics:

\{skip\}_m = m
Rules of Hoare Logic: Assignment

\[\vdash x := e : P \Rightarrow P[e/x]\]
Rules of Hoare Logic: Assignment

\[ \vdash x := e : P \Rightarrow P[e/x] \]

Is this correct?
Some instances

\[ x := x + 1 : \{ x < 0 \} \Rightarrow \{ x + 1 < 0 \} \]

Is this a valid triple?
Some instances

\( x := x + 1 : \{ x < 0 \} \Rightarrow \{ x + 1 < 0 \} \)

Is this a valid triple? \( \times \)
Some instances

\[ x := z + 1 : \{ x > 0 \} \Rightarrow \{ z + 1 > 0 \} \]

Is this a valid triple?
Some instances

\[ x := z + 1 : \{ x > 0 \} \Rightarrow \{ z + 1 > 0 \} \]

Is this a valid triple? ✗
Rules of Hoare Logic: Assignment

\[ \vdash x := e \quad : \quad P[e/x] \Rightarrow P \]
Rules of Hoare Logic: Assignment

\[ \text{Is this correct?} \]
Some instances

\[ x := z + 1 : \{ z + 1 > 0 \} \Rightarrow \{ x > 0 \} \]

Is this a valid triple?
Some instances

\[ x := z + 1 : \{ z + 1 > 0 \} \Rightarrow \{ x > 0 \} \]

Is this a valid triple? ✓
Some instances

\[ x := x + 1 : \{ x + 1 < 0 \} \Rightarrow \{ x < 0 \} \]

Is this a valid triple?
Some instances

\[ x := x + 1 : \{ x + 1 < 0 \} \Rightarrow \{ x < 0 \} \]

Is this a valid triple?  

✓
To show this rule correct we need to show the validity $x := e : P[e/x] \Rightarrow P$ for every $x, e, P$. 
Correctness Assignment Rule

\[ \vdash x := e : P[e/x] \Rightarrow P \]

To show this rule correct we need to show the validity \[ x := e : P[e/x] \Rightarrow P \] for every \( x, e, P \).

For every \( m \) such that \( P[e/x](m) \) and \( m' \) such that \( \{x := e\}_m = m' \) we need \( P(m') \).
Correctness Assignment Rule

\[ \vdash x := e : P[e/x] \Rightarrow P \]

To show this rule correct we need to show the validity \( x := e : P[e/x] \Rightarrow P \) for every \( x, e, P \).

For every \( m \) such that \( P[e/x](m) \) and \( m' \) such that \( \left\{ x := e \right\}_m = m' \) we need \( P(m') \).

By our semantics: \( \left\{ x := e \right\}_m = m[x = \{ e \}_m] \) and we can show \( P[e/x](m) = P(m[x = \{ e \}_m]) \).
Rules of Hoare Logic
Composition

⊢ c; c' : P ⊢ Q
Rules of Hoare Logic Composition

\[ \Gamma \vdash c : P \Rightarrow R \]

\[ \Gamma \vdash c ; c' : P \Rightarrow Q \]
Rules of Hoare Logic Composition

\[ \frac{\Gamma \vdash c : P \Rightarrow R \quad \Gamma \vdash c' : R \Rightarrow Q}{\Gamma \vdash c ; c' : P \Rightarrow Q} \]
Rules of Hoare Logic
Composition

\[ \vdash c : P \Rightarrow R \quad \vdash c' : R \Rightarrow Q \]

\[ \vdash c ; c' : P \Rightarrow Q \]

Is this correct?
Some Instances

⊢ \( x := z \cdot 2; \ z := x \cdot 2 \)

: \( (z \cdot 2) \cdot 2 = 8 \) \( \Rightarrow \) \( \{ z = 8 \} \)

Is this a valid triple?
Some Instances

⊢ \( x := z \times 2; z := x \times 2 \)

\[ (z \times 2) \times 2 = 8 \] \( \Rightarrow \) \( \{ z = 8 \} \)

Is this a valid triple? ✓
Some Instances

How can we prove it?

\[ \vdash x := z \ast 2; z := x \ast 2 : \{(z \ast 2) \ast 2 = 8\} \Rightarrow \{z = 8\} \]
Some Instances

How can we prove it?

\[
\begin{align*}
\vdash x \leftarrow z \times 2 : \{ (z \times 2) \times 2 = 8 \} & \Rightarrow \{ x \times 2 = 8 \} \\
\vdash z \leftarrow x \times 2 : \{ x \times 2 = 8 \} & \Rightarrow \{ z = 8 \} \\
\vdash x \leftarrow z \times 2; z \leftarrow x \times 2 : \{ (z \times 2) \times 2 = 8 \} & \Rightarrow \{ z = 8 \}
\end{align*}
\]
Correctness Composition Rule

\[ \vdash c : P \Rightarrow R \quad \vdash c' : R \Rightarrow Q \]

\[ \vdash c ; c' : P \Rightarrow Q \]

To show this rule correct we need to show the validity \( c ; c' : P \Rightarrow Q \) for every \( c, c', P, Q \).
Correctness Composition Rule

\[ \frac{\vdash c : P \Rightarrow R \quad \vdash c' : R \Rightarrow Q}{\vdash c ; c' : P \Rightarrow Q} \]

To show this rule correct we need to show the validity \( c ; c' : P \Rightarrow Q \) for every \( c, c', P, Q \).

For every \( m \) such that \( P(m) \) and \( m' \) such that \( \{ c, c' \}_m = m' \) we need \( Q(m') \).
Correctness Composition Rule

\[ \vdash c : P \Rightarrow R \quad \vdash c' : R \Rightarrow Q \]

\[ \vdash c ; c' : P \Rightarrow Q \]
Correctness Composition Rule

\[ \Gamma \vdash c : P \Rightarrow R \quad \Gamma \vdash c' : R \Rightarrow Q \]

\[ \Gamma \vdash c ; c' : P \Rightarrow Q \]

By our semantics: \( \{ c ; c' \} \_m = m' \) if and only if there is \( m'' \) such that \( \{ c \} \_m = m'' \) and \( \{ c' \} \_m'' = m' \).
Correctness Composition Rule

By our semantics: \( \{ c; c' \}_m = m' \) if and only if there is \( m'' \) such that \( \{ c \}_m = m'' \) and \( \{ c' \}_{m''} = m' \).

Assuming \( c: P \Rightarrow R \) and \( c': R \Rightarrow Q \) valid, if \( P(m) \) we can show \( R(m'') \) and if \( R(m'') \) we can show \( Q(m') \), hence since we have \( P(m) \) we can conclude \( Q(m') \).
Correctness Composition Rule

\[ \frac{\vdash c : P \Rightarrow R \quad \vdash c' : R \Rightarrow Q}{\vdash c ; c' : P \Rightarrow Q} \]

By our semantics: \( \{ c ; c' \}_m = m' \) if and only if there is \( m'' \) such that \( \{ c \}_m = m'' \) and \( \{ c' \}_{m''} = m' \).

Assuming \( c : P \Rightarrow R \) and \( c' : R \Rightarrow Q \) valid, if \( P(m) \) we can show \( R(m''') \) and if \( R(m''') \) we can show \( Q(m') \), hence since we have \( P(m) \) we can conclude \( Q(m') \). \( \checkmark \)
Is this a valid triple?
Some examples

\[ \vdash x := z \cdot 2; z := x \cdot 2 \]
\[ : \{ z \cdot 4 = 8 \} \Rightarrow \{ z = 8 \} \]

Is this a valid triple? ✓
Some examples

\[ \vdash x := z \cdot 2; z := x \cdot 2 \]

\[ \vdash \{ z \cdot 4 = 8 \} \Rightarrow \{ z = 8 \} \]

Is this a valid triple?

Can we prove it with the rules that we have?

✓
Some examples

\[ \vdash x := z \ast 2; \ z := x \ast 2 \]

: \{ z \ast 4 = 8 \} \Rightarrow \{ z = 8 \}

Is this a valid triple? \[\checkmark\]

Can we prove it with the rules that we have? \[\times\]
Some Instances

What is the issue?

\[ \vdash x := z \ast 2; z := x \ast 2 : \{ z \ast 4 = 8 \} \Rightarrow \{ z = 8 \} \]
Some Instances

What is the issue?

\[
\vdash x := z \times 2 : \{z \times 4 = 8\} \Rightarrow \{x \times 2 = 8\} \\
\vdash z := x \times 2 : \{x \times 2 = 8\} \Rightarrow \{z = 8\} \\
\vdash x := z \times 2; z := x \times 2 : \{z \times 4 = 8\} \Rightarrow \{z = 8\}
\]
Some Instances

What is the issue?

⊢ $x := z \times 2; \{z \times 4 = 8\} \Rightarrow \{x \times 2 = 8\}$

$\Rightarrow$

⊢ $z := x \times 2; \{x \times 2 = 8\} \Rightarrow \{z = 8\}$

⊢ $x := z \times 2; z := x \times 2; \{z \times 4 = 8\} \Rightarrow \{z = 8\}$

✗
Rules of Hoare Logic

Consequence

\[
\begin{align*}
P \Rightarrow S & \quad \vdash c : S \Rightarrow R & \quad \Rightarrow R \Rightarrow Q \\
\hline
\vdash c : P \Rightarrow Q
\end{align*}
\]
Some examples

\[ \vdash x := z * 2; z := x * 2 \]

: \{ z * 4 = 8 \} \Rightarrow \{ z = 8 \}

Is this a valid triple?
Some examples

\[ \vdash x := z \times 2; z := x \times 2 \]

: \{ z \times 4 = 8 \} \implies \{ z = 8 \}

Is this a valid triple? ✓
Some examples

\[ \vdash x := z \times 2; \ z := x \times 2 \]
\[ \implies \{ z \times 4 = 8 \} \Rightarrow \{ z = 8 \} \]

Is this a valid triple?

Can we prove it with the rules that we have?

✓
Some examples

<table>
<thead>
<tr>
<th>Is this a valid triple?</th>
<th>✓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can we prove it with the rules that we have?</td>
<td>✓</td>
</tr>
</tbody>
</table>
Some Instances

\[ \vdash x := z \cdot 2 \{ (z \cdot 2)^2 = 8 \} \Rightarrow \{ x \cdot 2 = 8 \} \]

\[ \{ z \cdot 4 = 8 \} \Rightarrow \{ (z \cdot 2)^2 = 8 \} \]

\[ \vdash x := z \cdot 2 : \{ z \cdot 4 = 8 \} \Rightarrow \{ x \cdot 2 = 8 \} \quad \vdash z := x \cdot 2 : \{ x \cdot 2 = 8 \} \Rightarrow \{ z = 8 \} \]

\[ \vdash x := z \cdot 2 ; z := x \cdot 2 : \{ z \cdot 4 = 8 \} \Rightarrow \{ z = 8 \} \]
Rules of Hoare Logic
If then else

⊢ if e then \(c_1\) else \(c_2\) : P \(\Rightarrow\) Q
Rules of Hoare Logic

If then else

\[ \vdash \text{if } e \text{ then } c_1 \text{ else } c_2 : P \Rightarrow Q \]

\[ \vdash c_1 : P \Rightarrow Q \quad \vdash c_2 : P \Rightarrow Q \]

\[ \vdash \text{if } e \text{ then } c_1 \text{ else } c_2 : P \Rightarrow Q \]
Rules of Hoare Logic
If then else

\[
\begin{align*}
\vdash c_1 : P \Rightarrow Q & \quad \vdash c_2 : P \Rightarrow Q \\
\hline
\vdash \text{if } e \text{ then } c_1 \text{ else } c_2 : P \Rightarrow Q
\end{align*}
\]

Is this correct?
Some examples

⊢ if y = 0 then skip else x := x + 1; x := x − 1

: \{ x = 1 \} \Rightarrow \{ x = 1 \}

Is this a valid triple?
Some examples

⊢ if y = 0 then skip else x := x + 1; x := x − 1

: {x = 1} ⇒ {x = 1}

Is this a valid triple? ✓
Some examples

⊢ if y = 0 then skip else x := x + 1; x := x − 1

: {x = 1} ⇒ {x = 1}

Is this a valid triple? ✓

Can we prove it with the rules that we have?
Some examples

⊢ if \( y = 0 \) then skip else \( x := x + 1 ; x := x - 1 \)

\[ : \{ x = 1 \} \Rightarrow \{ x = 1 \} \]

Is this a valid triple? ✔

Can we prove it with the rules that we have? ✔
Some Instances

\[ \text{\textbf{⊢}} \text{uskip: } \{x = 1\} \Rightarrow \{x = 1\} \quad \text{\textbf{⊢}} x := x + 1; x := x - 1 : \{x = 1\} \Rightarrow \{x = 1\} \]

\[ \quad \text{\textbf{⊢}} \text{if } y = 0 \text{ then skip else } x := x + 1; x := x - 1 \]

\[ : \{x = 1\} \Rightarrow \{x = 1\} \]
Rules of Hoare Logic
If then else

\[
\vdash c_1 : P \Rightarrow Q \quad \vdash c_2 : P \Rightarrow Q
\]

\[
\vdash \text{if } e \text{ then } c_1 \text{ else } c_2 : P \Rightarrow Q
\]
Rules of Hoare Logic
If then else

\[ \vdash c_1 : P \Rightarrow Q \quad \vdash c_2 : P \Rightarrow Q \]

\[ \vdash \text{if } e \text{ then } c_1 \text{ else } c_2 : P \Rightarrow Q \]

Is this strong enough?
Some examples

\[
\begin{array}{c}
\downarrow \text{ if false then skip else } x = x + 1 \\
: \{ x = 0 \} \Rightarrow \{ x = 1 \}
\end{array}
\]

Is this a valid triple?
Some examples

⊢ if false then skip else $x = x + 1$

: $\{x = 0\} \Rightarrow \{x = 1\}$

Is this a valid triple? ✓
Some examples

\[ \text{⊢ if false then skip else } x = x + 1 \]
\[ : \{ x = 0 \} \Rightarrow \{ x = 1 \} \]

Is this a valid triple?  

Can we prove it with the rules that we have?  

✓
Some examples

⊢ if false then skip else \( x = x + 1 \) : \{ x = 0 \} \Rightarrow \{ x = 1 \}

Is this a valid triple? ✓

Can we prove it with the rules that we have? ✗
Rules of Hoare Logic
If then else

\[ \vdash \text{if } e \text{ then } c_1 \text{ else } c_2 : P \Rightarrow Q \]

Is this correct?
Rules of Hoare Logic
If then else

\[ \vdash \text{if } e \text{ then } c_1 \text{ else } c_2 : P \Rightarrow Q \]

\[ \vdash c_1 : e \land P \Rightarrow Q \quad \vdash c_2 : \neg e \land P \Rightarrow Q \]

\[ \text{Is this correct?} \]

\[ \text{Homework} \]
Rules of Hoare Logic: Abort

\[ \vdash \text{Abort}: \ ? \Rightarrow ? \]
Rules of Hoare Logic: Abort

\[ \vdash \text{Abort: } ? \Rightarrow ? \]

What can be a good specification?
Validity of Hoare triple

We say that the triple $c : P \Rightarrow Q$ is valid if and only if for every memory $m$ such that $P(m)$ and memory $m'$ such that $\{c\}_m = m'$ we have $Q(m')$. 
Rules of Hoare Logic:

\[
\text{Abort} \quad \vdash \quad \text{Abort} : P \Rightarrow Q
\]
Rules of Hoare Logic: Abort

\[ \Box \text{Abort} : \Box P \Rightarrow Q \]

To show this rule correct we need to show the validity \( \Box \text{Abort} : P \Rightarrow Q \) for every \( P, Q \).
Rules of Hoare Logic:

**Abort**

\[ \vdash \text{Abort} : P \Rightarrow Q \]

To show this rule **correct** we need to show the **validity** \( \text{Abort} : P \Rightarrow Q \) for every \( P, Q \).

For every \( m \) such that \( P(m) \) and \( m' \) such that \( \{ \text{Abort} \}_{m=m'} \) we need \( Q(m') \).
Rules of Hoare Logic: Abort

\[ \vdash \text{Abort} : P \rightarrow Q \]

To show this rule correct we need to show the validity \( \text{Abort} : P \rightarrow Q \) for every \( P, Q \).

For every \( m \) such that \( P(m) \) and \( m' \) such that \( \{ \text{Abort} \}_{m=m'} \) we need \( Q(m') \).

Vacuously True
Rules of Hoare Logic

While

\[ \vdash \text{while } e \text{ do } c : \text{ ??} \]
Rules of Hoare Logic

While

\[ \neg e \Rightarrow P \]

\[ \text{while } e \text{ do } c : P \Rightarrow P \]
Rules of Hoare Logic

While

\[ \frac{P \Rightarrow e \quad \vdash c : P \Rightarrow P}{\vdash \text{while } e \text{ do } c : P \Rightarrow P} \]
Rules of Hoare Logic

While

$\vdash c : e \land P \Rightarrow P$

$\vdash \text{while } e \text{ do } c : P \Rightarrow P \land \neg e$

Invariant
An example

\[ \text{while } x = 0 \text{ do } x := x + 1 \]

: \{ x = 1 \} \Rightarrow \{ x = 1 \}

How can we derive this?
An example

⊢ while \( x = 0 \) do \( x := x + 1 \)

\[ : \{ x = 1 \} \Rightarrow \{ x = 1 \} \]

What can be a good Invariant?
An example

\[ \text{\texttt{\textbf{\textcolor{blue}{\textbf{while}}} \quad x = 0 \quad \texttt{\textbf{do}} \quad x := x + 1} \]

\[ : \{x = 1\} \Rightarrow \{x = 1\} \]

What can be a good Invariant?

\[ Inv = \{x = 1\} \]
An example

\[ \text{⊢ } \frac{\text{while } x = 0 \text{ do } x := x + 1: \{x = 1\}}{\{x = 1\}} \]
An example

\[ \vdash \text{while } x = 0 \text{ do } x := x + 1: \{ x = 1 \} \Rightarrow \{ x = 1 \land x \neq 0 \} \quad x = 1 \land x \neq 0 \Rightarrow x = 1 \]

\[ \vdash \text{while } x = 0 \text{ do } x := x + 1: \{ x = 1 \} \Rightarrow \{ x = 1 \} \]
An example

\[ x = 1 \land x = 0 \Rightarrow x + 1 = 1 \]

\[ \vdash x := x + 1 : \{ x + 1 = 1 \} \Rightarrow \{ x = 1 \} \]

\[ \vdash x := x + 1 : \{ x = 1 \land x = 0 \} \Rightarrow \{ x = 1 \} \]

\[ \vdash \text{while } x = 0 \text{ do } x := x + 1 : \{ x = 1 \} \Rightarrow \{ x = 1 \land x \neq 0 \} \]

\[ x = 1 \land x \neq 0 \Rightarrow x = 1 \]

\[ \vdash \text{while } x = 0 \text{ do } x := x + 1 : \{ x = 1 \} \Rightarrow \{ x = 1 \} \]
An example

\[ x = 1 \land x = 0 \Rightarrow x + 1 = 1 \]

\[ \vdash x := x + 1 : \{ x + 1 = 1 \} \Rightarrow \{ x = 1 \} \]

\[ \vdash x := x + 1 : \{ x = 1 \land x = 0 \} \Rightarrow \{ x = 1 \} \]

\[ \vdash \text{while } x = 0 \text{ do } x := x + 1 : \{ x = 1 \} \Rightarrow \{ x = 1 \land x \neq 0 \} \]

\[ x = 1 \land x \neq 0 \Rightarrow x = 1 \]

\[ \vdash \text{while } x = 0 \text{ do } x := x + 1 : \{ x = 1 \} \Rightarrow \{ x = 1 \} \]
Another example

How can we derive this?

\[
\begin{array}{l}
x := 3; \\
y := 1; \\
\text{while } x > 1 \text{ do} \\
y := y + 1; \\
x := x - 1;
\end{array}
\]

\[\vdash \{ \text{true} \} \Rightarrow \{ y = 3 \} \]
Another example

\begin{align*}
\text{x:=3;} \\
y:=1; \\
\text{while } x > 1 \text{ do} \\
\quad y := y+1; \\
\quad x := x-1;
\end{align*}

\therefore \quad \{true\} \Rightarrow \{y = 3\}

What can be a good Invariant?
Another example

```plaintext
x := 3;
y := 1;
while x > 1 do
  y := y + 1;
x := x - 1;
```

⊢ : \{true\} ⇒ \{y = 3\}

What can be a good Invariant?

\[ \text{Inv} = \{y = 4 - x \land x \geq 1\} \]
How do we know that these are the right rules?
Soundness

If we can derive $\vdash c : P \Rightarrow Q$ through the rules of the logic, then the triple $c : P \Rightarrow Q$ is valid.
Are the rules we presented sound?
Completeness

If a triple $c : P \Rightarrow Q$ is valid, then we can derive $\vdash c : P \Rightarrow Q$ through the rules of the logic.
Are the rules we presented complete?
Relative Completeness

\[ P \Rightarrow S \quad \vdash c : S \Rightarrow R \quad R \Rightarrow Q \]

\[ \vdash c : P \Rightarrow Q \]
Relative Completeness

\[ P \Rightarrow S \quad \vdash c : S \Rightarrow R \quad R \Rightarrow Q \]

\[ \vdash c : P \Rightarrow Q \]

If a triple \( c : \text{Pre} \Rightarrow \text{Post} \) is valid, and we have an oracle to derive all the true statements of the form \( P \Rightarrow S \) and of the form \( R \Rightarrow Q \), which we can use in applications of the conseq rule, then we can derive \( \vdash c : \text{Pre} \Rightarrow \text{Post} \) through the rules of the logic.