CS 591: Formal Methods in Security and Privacy
Hoare Logic

Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu
LfA
Formal Semantics

We need to assign a formal meaning to the different components:

- **Precondition**
- **Program**
- **Postcondition**

We also need to describe the rules which combine program and specifications.

**Formal semantics of programs**

**Formal semantics of specification conditions**
Semantics of Commands

This is defined on the structure of commands:

\[ \{\text{abort}\}_m = \bot \]
\[ \{\text{skip}\}_m = m \]
\[ \{x:=e\}_m = m[x\leftarrow \{e\}_m] \]
\[ \{c; c'\}_m = \{c'\}_{m'} \quad \text{if} \quad \{c\}_m = m' \]
\[ \{c; c'\}_m = \bot \quad \text{if} \quad \{c\}_m = \bot \]
\[ \{\text{if } e \text{ then } c_t \text{ else } c_f\}_m = \{c_t\}_m \quad \text{if } \{e\}_m = \text{true} \]
\[ \{\text{if } e \text{ then } c_t \text{ else } c_f\}_m = \{c_f\}_m \quad \text{if } \{e\}_m = \text{false} \]
\[ \{\text{while } e \text{ do } c\}_m = \sup_{n \in \mathbb{N}} \{\text{while}_n e \text{ do } c\}_m \]

where

\[ \text{while}_n e \text{ do } c = \text{while}_n e \text{ do } c; \text{if } e \text{ then abort else skip} \]
Semantics of Commands

This is defined on the structure of commands:

\[
\begin{align*}
\{\text{abort}\}_m &= \bot \\
\{\text{skip}\}_m &= m \\
\{x := e\}_m &= m[x \leftarrow \{e\}_m] \\
\{c;c'\}_m &= \{c'\}_m' & \text{if } & \{c\}_m = m' \\
\{c;c'\}_m &= \bot & \text{if } & \{c\}_m = \bot \\
\{\text{if } e \text{ then } c_t \text{ else } c_f\}_m &= \{c_t\}_m & \text{if } & \{e\}_m = \text{true} \\
\{\text{if } e \text{ then } c_t \text{ else } c_f\}_m &= \{c_f\}_m & \text{if } & \{e\}_m = \text{false} \\
\{\text{while } e \text{ do } c\}_m &= \sup_{n \in \mathbb{N}} \{\text{while}_n e \text{ do } c\}_m \\
\text{where} & \\
\text{while}_n e \text{ do } c &= \text{while}_n e \text{ do } c; \text{if } e \text{ then } \text{abort } \text{ else } \text{skip} \\
\text{and} & \\
\text{while}_0 e \text{ do } c &= \text{skip} \\
\text{while}_{n+1} e \text{ do } c &= \text{if } e \text{ then } (c; \text{while}_n e \text{ do } c) \text{ else } \text{skip}
\end{align*}
\]
Program Specifications (Hoare Triples)
Specifications - Hoare triple

Precondition
Program
Postcondition

\[ c : P \implies Q \]

Program

Postcondition (a logical formula)
Some examples

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

Is it a good specification?
Some examples

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

Is it a good specification?

✓
Specification can also be imprecise.
Some examples

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

Precondition

Is it a good specification?

Postcondition
Some examples

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

Is it a good specification?

✓
Some examples

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

Is it a good specification?
Some examples

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

Is it a good specification?

✓
Some examples

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

Is it a good specification?
Some examples

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

Is it a good specification? ✓
Some examples

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

**Precondition**

**Postcondition**

Is it a good specification?  \( \times \)

\[ m_{in} = [z = -1, x = 2] \quad m_{out} = [z = -1, x = 0] \]
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

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

Postcondition

Is it a good specification?
Some examples

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

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

Postcondition

Is it a good specification?
Some examples

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

\[
\begin{align*}
: \{0 \leq k\} & \Rightarrow \{r = n^k\}
\end{align*}
\]

Is it a good specification?

\[
\begin{align*}
m_{in} & = [k = 0,n = 2,i = 0,r = 0] \\
m_{out} & = [k = 0,n = 2,i = 1,r = 2]
\end{align*}
\]
Some examples

Precondition

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

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

Postcondition

Is it a good specification?
Some examples

Precondition

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

Postcondition

Is it a good specification?

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

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

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

Precondition

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

Postcondition

Is it a good specification?

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

Precondition

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

Postcondition

Is it a good specification?

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

Is it a good specification?
Some examples

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

Postcondition

Is it a good specification?

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

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

Is it a good specification?

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

- **Postcondition**: 
  \[ \text{Is it a good specification?} \]
Some examples

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

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

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?

✓

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

E : P ⇒ 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 $Q$.

$P \Rightarrow Q$
Validity of Hoare triple

Precondition (a logical formula) \( c : P \Rightarrow Q \) Postcondition (a logical formula)

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

Robert W Floyd

Tony Hoare

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

\[ \vdash \text{skip}: P \Rightarrow P \]
Rules of Hoare Logic: Skip

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

Is this correct?
Correctness of an axiom

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

⊢ skip: P ⇒ P

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

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

To show this rule correct we need to show the validity of the triple \( \text{skip: } P \Rightarrow P \).

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

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

To show this rule correct we need to show the validity of the triple \( \text{skip}: P \Rightarrow P \).

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

Follow easily by our semantics:

\[ \{\text{skip}\}_m = m \]
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? ✗
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

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

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

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? Yes

\[ \checkmark \]
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? ✓
Correctness Assignment Rule

\[ \vdash x := \textbf{e} : P[e/x] \Rightarrow P \]

To show this rule correct we need to show the validity \( x := \textbf{e} : P[e/x] \Rightarrow P \) for every \( x, \textbf{e}, P \).
To show this rule correct we need to show the validity $\frac{x := e : P[e/x] \Rightarrow P}{\text{for every } x, e, P}$. 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 \( \{x := e\}_m = m' \) we need \( P(m') \).

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

Composition

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

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

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

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

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

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

\[ \vdash c ; c' : \text{P} \Rightarrow Q \]

Is this correct?
Some Instances

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

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

Is this a valid triple?
Some Instances

\[ \begin{align*}
\vdash \ & x := z \ast 2; \ z := x \ast 2 \\
\therefore \ & ((z \ast 2) \ast 2 = 8) \ \Rightarrow \ \{ z = 8 \}
\end{align*} \]

Is this a valid triple?  

✓
Some Instances

How can we prove it?

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

How can we prove it?

\[
\begin{align*}
\vdash x & := z \times 2 : \{(z \times 2) \times 2 = 8\} \Rightarrow \{x \times 2 = 8\} \\
\vdash z & := x \times 2 : \{x \times 2 = 8\} \Rightarrow \{z = 8\} \\
\vdash x & := z \times 2; z := x \times 2 : \{(z \times 2) \times 2 = 8\} \Rightarrow \{z = 8\}
\end{align*}
\]
To show this rule correct we need to show the validity $c; c' : P \Rightarrow Q$ for every $c, c', P, Q$. 

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

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

$$\Gamma c : P \Rightarrow R \quad \Gamma c' : R \Rightarrow Q$$

$$\Gamma c ; c' : P \Rightarrow Q$$
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 \( \{ c' \}_{m''} = m' \).
Correctness Composition Rule

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

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

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

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

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

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

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

\( \{ c \}_m = m'' \) and \( \{ c' \}_m = m' \).

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 \( Q(m') \).
Some examples

⊢ $x := z \ast 2; z := x \ast 2$

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

Is this a valid triple?
Some examples

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

Is this a valid triple?  ✓
Some examples

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

\[ : \{ z \times 4 = 8 \} \Rightarrow \{ z = 8 \} \]

Is this a valid triple?

Can we prove it with the rules that we have?

✓
Some examples

\[ \begin{align*}
\vdash x & := z * 2; z := x * 2 \\
& : \{ z * 4 = 8 \} \Rightarrow \{ z = 8 \}
\end{align*} \]

Is this a valid triple? \(\checkmark\)

Can we prove it with the rules that we have? \(\times\)
Some Instances

What is the issue?

$\vdash x := z \cdot 2; z := x \cdot 2 : \{z \cdot 4 = 8\} \Rightarrow \{z = 8\}$
Some Instances

What is the issue?

\[
\begin{align*}
\vdash & x := z * 2 : \{z \times 4 = 8\} \Rightarrow \{x \times 2 = 8\} \\
\vdash & z := x \times 2 : \{x \times 2 = 8\} \Rightarrow \{z = 8\} \\
\vdash & x := z \times 2; z := x \times 2 : \{z \times 4 = 8\} \Rightarrow \{z = 8\}
\end{align*}
\]
Some Instances

What is the issue?

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

\[\Gamma \vdash z := x * 2; \{x * 2 = 8\} \Rightarrow \{z = 8\}\]

\[\Gamma \vdash x := z * 2; z := x * 2; \{z * 4 = 8\} \Rightarrow \{z = 8\}\]
Rules of Hoare Logic

Consequence

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

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

⊢ $x := z \ast 2; z := x \ast 2$

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

Is this a valid triple?  ✔️
Some examples

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

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

Is this a valid triple? ✓

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

⊢ $x := z \cdot 2; z := x \cdot 2$

: {$z \cdot 4 = 8} \Rightarrow \{z = 8\}$

Is this a valid triple? ✓

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

\[
\begin{align*}
\vdash x &:= z \cdot 2 \{ (z \cdot 2) \cdot 2 = 8 \} \Rightarrow \{ x \cdot 2 = 8 \} \\
\{ z \cdot 4 = 8 \} &\Rightarrow \{ (z \cdot 2) \cdot 2 = 8 \} \\
\vdash x &:= z \cdot 2: \{ z \cdot 4 = 8 \} \Rightarrow \{ x \cdot 2 = 8 \} \\
\vdash z &:= x \cdot 2: \{ x \cdot 2 = 8 \} \Rightarrow \{ z = 8 \} \\
\vdash x &:= z \cdot 2; z := x \cdot 2: \{ z \cdot 4 = 8 \} \Rightarrow \{ z = 8 \}
\end{align*}
\]
Rules of Hoare Logic
If then else

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

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

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

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

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

Is this correct?
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 \( x := x + 1; x := x - 1 \)

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

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? ✓

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

\[
\begin{array}{l}
\vdash \text{skip}: \{x = 1\} \Rightarrow \{x = 1\} \\
\vdash x := x + 1; x := x - 1 : \{x = 1\} \Rightarrow \{x = 1\} \\
\vdash \text{if } y = 0 \text{ then skip else } x := x + 1; x := x - 1 \\
\phantom{\vdash} : \{x = 1\} \Rightarrow \{x = 1\}
\end{array}
\]
Rules of Hoare Logic
If then else

\[
\begin{array}{c}
\vdash c_1 : P \Rightarrow Q \\
\hline
\vdash c_2 : P \Rightarrow Q \\
\hline
\vdash \text{if } e \text{ then } c_1 \text{ else } c_2 : P \Rightarrow Q
\end{array}
\]
Rules of Hoare Logic

If then else

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

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

Is this strong enough?
Some examples

⊢ if false then skip else \( x = x + 1 \)

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

Is this a valid triple?
Some examples

⊢ if false then skip else \( x = x + 1 \)

: \{ x = 0 \} \implies \{ x = 1 \}

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? ✓

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

\[ \text{Is this a valid triple?} \quad \checkmark \]

\[ \text{Can we prove it with the rules that we have?} \quad \times \]

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

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

Is this correct?

Homework
Rules of Hoare Logic: Abort

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

⊢ Abort: ? ⇒ ?

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:
Abort

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

Is this correct?
Rules of Hoare Logic: Abort

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

Is this correct?

Homework