CS 591: Formal Methods in Security and Privacy
More Hoare Logic

Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu
From the previous classes
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.
Programming Language

\[
c ::= \text{abort} \\
| \text{skip} \\
| x := e \\
| c ; c \\
| \text{if } e \text{ then } c \text{ else } c \\
| \text{while } e \text{ do } c
\]

\[x, y, z, \ldots\] program variables

\[e_1, e_2, \ldots\] expressions

\[c_1, c_2, \ldots\] commands
Summary of the Semantics 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} \quad \{e\}_m = \text{true}

\{\text{if } e \text{ then } c_t \text{ else } c_f\}_m = \{c_f\}_m \quad \text{if} \quad \{e\}_m = \text{false}

\{\text{while } e \text{ do } c\}_m = \sup_{n \in \mathbb{Nat}}\{\text{while}_n e \text{ do } c\}_m
Hoare triple

$c : P \Rightarrow Q$
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?
Rules of Hoare Logic
Skip

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

Assignment

\[ \vdash x := e \quad : \quad P \left[ e / x \right] \implies P \]
Rules of Hoare Logic Composition

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

\[ \vdash c; c' : P \Rightarrow Q \]
We can \textit{weaken} \( P \), i.e. replace it by something that is implied by \( P \). In this case \( S \).

We can \textit{strengthen} \( Q \), i.e. replace it by something that implies \( Q \). In this case \( R \).
Today: More Hoare Logic
Rules of Hoare Logic
If then else

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

Is this correct?
Correctness of a rule

We say that a rule is **correct** if given **valid triples** as described by the assumption(s), we can prove the **validity of the triple** in the conclusion.
Rules of Hoare Logic

If then else

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

Is this correct?
Rules of Hoare Logic
If then else

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

Is this strong enough?
Some examples

\[
\begin{align*}
\text{\footnotesize \{x = 1\}} & \implies \text{\footnotesize \{x = 1\}} \\
\text{if true then skip else } x &= x + 1
\end{align*}
\]

How can we derive this?
Rules of Hoare Logic
If then else

\[ \vdash_c e \land P \Rightarrow Q \quad \vdash_c \neg e \land P \Rightarrow Q \]

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

While

\[ \vdash c : \ ?? \]

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

While

\[ \vdash c : e \land P \implies P \]

\[ \vdash \text{while } e \text{ do } c : P \implies P \land \neg e \]

Invariant
Some examples

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

How can we derive this?
Some examples

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

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

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

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

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

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

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

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

How can we derive this?