CS 839 Systems Verification
Lecture 6: Hoare logic (part 2)

this lecture benefits from projecting the slides

Learning outcomes

1. Prove reasoning principles in Hoare logic
2. Analyze pre- and post-conditions

Quiz: what is soundness?

$\{P\}\, e\, \{\lambda v.\, Q(v)\}$

$e \rightarrow^* e'$ execution relation

5 MIN

Answer

$$\{P\}\, e\, \{\lambda v.\, Q(v)\}$$

$$\forall v', P \wedge (e \rightarrow^* v') \implies Q(v')$$

Other "soundness" definitions

Task: commit to reasonable or not, then discuss in pairs the ones you disagree on

1. $\forall v', P \wedge (e \to^* v') \implies Q(v')$ *(original)*
2. $P \implies \exists v', e \to^* v' \wedge Q(v')$
3. $P \wedge (\forall v', e \to^* v' \implies Q(v'))$
4. $P \implies (\exists v', e \to^* v') \wedge (\forall v', e \to^* v' \implies Q(v'))$
5. $\exists v', (P \wedge e \to^* v') \implies Q(v')$

10 MIN for think-pair, **10 MIN** for debrief

commit to reasonable/not reasonable

discuss which ones you disagree on

Answers:

original definition
one path is correct
nonsense: says precondition holds and postcondition holds unconditionally
total correctness
nonsense: always true (says there exists such that an implication holds; if the exists makes the left-hand side of the implication false, automatically holds)

4 definitely reasonable, 2 is probably not, 3 and 5 definitely not

5-min break

Proof system

$$\frac{\{P\}\ e_1\ \{\lambda v.\, Q(v)\} \quad \forall v.\ \{Q(v)\}\ e_2[v/x]\ \{R\}}{\{P\}\ \textbf{let}\ x := e_1\ \textbf{in}\ e_2\ \{R\}}\ \text{hoare-let}$$

Example: verify directly against soundness

5 MIN

Exercise: Rule of consequence

prove this rule from the definition of soundness

$$\frac{P' \vdash P \quad (\forall v.\, Q(v) \vdash Q'(v)) \quad \{P\}\ e\ \{Q(v)\}}{\{P'\}\ e\ \{\lambda v.\, Q'(v)\}}\ \text{consequence}$$

10 MIN (think-pair, whole group discussion)

Bonus exercise: prove pure step

$$\frac{e_1 \to e_2 \quad \{P\}\ e_2\ \{\lambda v.\, Q(v)\}}{\{P\}\ e_1\ \{\lambda v.\, Q(v)\}}\ \text{pure-step}$$

Need determinism as a lemma, but then the rule makes sense

Example specs

$$\text{and} = \lambda b_1, b_2.\, \textbf{if}\ b_1\ \textbf{then}\ b_2\ \textbf{else}\ \text{false}$$
$$\text{add} = \lambda x, y.\, x + y$$
$$\text{min} = \lambda x, y.\, \textbf{if}\ x < y\ \textbf{then}\ x\ \textbf{else}\ y$$

$$\{\text{True}\}\ \text{and}\ b_1\, b_2\ \{\lambda v.\, v = \overline{b_1 \,\&\, b_2}\}$$
$$\{n + m < 2^{64}\}\ \text{add}\ \overline{n}\, \overline{m}\ \{\lambda v.\, v = \overline{n + m}\}$$
$$\{\text{True}\}\ \text{min}\ \overline{n}\, \overline{m}\ \{\lambda v.\, \exists (p : \mathbb{Z}).\, v = \overline{p} \land p \le n \land p \le m\}$$

Things to note: and has a reasonably strong specification, add has a too-strong precondition, min has an under-specified postcondition

5 MIN

Exercise: alternate specifications

1. What is a stronger specification for `min`?
2. Can you generalize the spec for `add`?
3. Can you generalize the spec for `and`? (tricky)

$$\text{and} = \lambda b_1, b_2.\, \textbf{if}\; b_1 \;\textbf{then}\; b_2 \;\textbf{else}\; \text{false}$$

$$\text{add} = \lambda x, y.\, x + y$$

$$\text{min} = \lambda x, y.\, \textbf{if}\; x < y \;\textbf{then}\; x \;\textbf{else}\; y$$

$$\{\text{True}\}\ \text{and}\ b_1\, b_2\ \{\lambda v.\, v = \overline{b_1\ \&\ b_2}\}$$

$$\{n + m < 2^{64}\}\ \text{add}\ \overline{n}\,\overline{m}\ \{\lambda v.\, v = \overline{n + m}\}$$

$$\{\text{True}\}\ \text{min}\ \overline{n}\,\overline{m}\ \{\lambda v.\, \exists (p : \mathbb{Z}).\, v = \overline{p} \wedge p \leq n \wedge p \leq m\}$$

10 MIN

Verifying a function

$f = \lambda x.\, \mathrm{add}\,(\min 0\, x)\, 1$

$\{n < 2^{64} - 1\}$

$\quad f\, \overline{n}$

$\{\lambda v.\, \exists (p : \mathcal{Z}).\, v = \overline{p} \wedge p \leq 1\}$

Recall: rule of consequence

$$\frac{P' \vdash P \quad (\forall v.\, Q(v) \vdash Q'(v)) \quad \{P\}\ e\ \{Q(v)\}}{\{P'\}\ e\ \{\lambda v.\, Q'(v)\}}\ \text{consequence}$$

This rule is important for adapting Hoare triples as needed. Allow us to prove the strongest specification we care to and then keep using it, without having to revisit that *proof*.

Proof outlines

$$
\begin{array}{l}
\{n < 2^{64} - 1\} \\
\{\text{True}\} \\
\quad \textbf{let } m := \text{min } 0\, \overline{n} \textbf{ in} \\
\{\exists p_m.\, m = \overline{p_m} \land p_m \leq 0 \land p_m \leq n\} \\
\{\overline{m} + 1 < 2^{64}\} \\
\quad \textbf{let } y := \text{add } m\, 1 \textbf{ in} \\
\{y = \overline{m + 1}\} \\
\quad y \\
\{y = \overline{p_m + 1} \land p_m + 1 \leq 1\} \\
\{\exists (p : \mathbb{Z}).\, y = \overline{p} \land p \leq 1)\}
\end{array}
$$

Need to recall our (under-specified) min spec and our add spec

15 MIN

Better soundness

$\{P\}\, e\, \{v.\, Q(v)\} \triangleq$

If P holds and $e \to^* e'$, either

- (a) e' is not stuck OR
- (b) there is a value v' $e' = v'$ and $Q(v')$ holds.