COMP3610/6361
Principles of Programming Languages

Peter Höfner

Sep 27, 2023

Section 17

Axiomatic Semantics

Floyd-Hoare Logic

Idea: develop proof system as an inductively-defined set; every member will be a valid partial correctness statement

Judgement

$$\vdash \{P\}\ c\ \{Q\}$$

Floyd-Hoare Logic – Skip

(skip) $\vdash \{P\}$ **skip** $\{P\}$

Floyd-Hoare Logic – Assignment

(assign) $\vdash \{P[a/l]\}\ l := a\ \{P\}$

Notation: $P[a/l]$ denotes substitution of a for l in P;
in operational semantics we wrote $\{a/l\}\,P$

Example

$$\{7 = 7\}\ l := 7\ \{l = 7\}$$

Australian National University

Floyd-Hoare Logic – Incorrect Assignment

(wrong1) $\vdash \{P\}\; l := a\; \{P[a/l]\}$

Example

$$\{l = 0\}\; l := 7\; \{7 = 0\}$$

(wrong2) $\vdash \{P\}\; l := a\; \{P[l/a]\}$

Example

$$\{l = 0\}\; l := 7\; \{l = 0\}$$

Floyd-Hoare Logic – Sequence, If, While

(seq)
$$\frac{\vdash \{P\}\ c_1\ \{R\} \qquad \vdash \{R\}\ c_2\ \{Q\}}{\vdash \{P\}\ c_1\ ;\ c_2\ \{Q\}}$$

(if)
$$\frac{\vdash \{P \wedge b\}\ c_1\ \{Q\} \qquad \vdash \{P \wedge \neg b\}\ c_2\ \{Q\}}{\vdash \{P\}\ \textbf{if}\ b\ \textbf{then}\ c_1\ \textbf{else}\ c_2\ \{Q\}}$$

(while)
$$\frac{\vdash \{P \wedge b\}\ c\ \{P\}}{\vdash \{P\}\ \textbf{while}\ b\ \textbf{do}\ c\ \{P \wedge \neg b\}}$$

P acts as *loop invariant*

Floyd-Hoare Logic – Consequence

We cannot combine arbitrary triple yet

$$\dfrac{\dfrac{}{\vdash \{3 = 3\}\ l := 3\ \{l = 3\}}\text{(assign)} \qquad \dfrac{\dots}{\vdash \{l \geq 2\}\ l :=!l - 2\ \{l \geq 0\}}}{\vdash \{3 = 3\}\ l := 3\ ;\ l :=!l - 2\ \{l \geq 0\}}$$

Floyd-Hoare Logic – Consequence

strengthen pre-conditions and weaken post-conditions

$$\text{(cons)} \quad \frac{\models P \Rightarrow P' \qquad \vdash \{P'\}\, c\, \{Q'\} \qquad \models Q' \Rightarrow Q}{\vdash \{P\}\, c\, \{Q\}}$$

Recall: $\models P \Rightarrow P'$ denotes assertion validity

Floyd-Hoare Logic – Summary

(skip) $\vdash \{P\}\ \textbf{skip}\ \{P\}$

(assign) $\vdash \{P[a/l]\}\ l := a\ \{P\}$

(seq) $$\frac{\vdash \{P\}\ c_1\ \{R\} \qquad \vdash \{R\}\ c_2\ \{Q\}}{\vdash \{P\}\ c_1\ ;\ c_2\ \{Q\}}$$

(if) $$\frac{\vdash \{P \wedge b\}\ c_1\ \{Q\} \qquad \vdash \{P \wedge \neg b\}\ c_2\ \{Q\}}{\vdash \{P\}\ \textbf{if}\ b\ \textbf{then}\ c_1\ \textbf{else}\ c_2\ \{Q\}}$$

(while) $$\frac{\vdash \{P \wedge b\}\ c\ \{P\}}{\vdash \{P\}\ \textbf{while}\ b\ \textbf{do}\ c\ \{P \wedge \neg b\}}$$

(cons) $$\frac{\models P \Rightarrow P' \qquad \vdash \{P'\}\ c\ \{Q'\} \qquad \models Q' \Rightarrow Q}{\vdash \{P\}\ c\ \{Q\}}$$

Floyd-Hoare Logic – Exercise

$\{l_0 = n \wedge n > 0\}$

$l_1 := 1$;

while $!l_0 > 0$ **do**

$\quad l_1 := !l_1 \cdot !l_0$;

$\quad l_0 := !l_0 - 1$

$\{l_1 = n!\}$

Soundness and Completeness

how do $\vdash$ (judgement) and $\models$ (validity) relate?

Soundness:
if a partial correctness statement can be derived ($\vdash$) then is is valid ($\models$)

Completeness:
if the statement is valid ($\models$) then a derivation exists ($\vdash$)

Soundness and Completeness

Theorem (Soundness)

If $\vdash \{P\}\ c\ \{Q\}$ *then* $\models \{P\}\ c\ \{Q\}$.

Proof.

Induction on the derivation of $\vdash \{P\}\ c\ \{Q\}$. $\square$

Soundness and Completeness

Conjecture (Completeness)

If $\models \{P\}\ c\ \{Q\}$ *then* $\vdash \{P\}\ c\ \{Q\}$.

Rule (cons) spoils completeness

$$\text{(cons)} \quad \frac{\models P \Rightarrow P' \qquad \vdash \{P'\}\ c\ \{Q'\} \qquad \models Q' \Rightarrow Q}{\vdash \{P\}\ c\ \{Q\}}$$

Can we derive $\models P \Rightarrow P'$?
No, according to Gödel's incompleteness theorem (1931)

Soundness and Completeness

Theorem (Relative Completeness)

$P, Q \in$ *assn*, $c \in$ *com*. $\models \{P\}\ c\ \{Q\}$ *implies* $\vdash \{P\}\ c\ \{Q\}$.

Floyd-Hoare logic is no more incomplete than our language of assertions

Proof depends on the notion of *weakest liberal preconditions*.

Decorated Programs

Observation: once loop invariants and uses of consequence are identified, the structure of a derivation in Floyd-Hoare logic is determined
Write "proofs" by decorating programs with:

- a precondition ($\{P\}$)
- a postcondition ($\{Q\}$)
- invariants ($\{I\}$**while** b **do** c)
- uses of consequence ($\{R\} \Rightarrow \{S\}$)
- assertions between sequences (c_1 ; $\{T\}c_2$)

decorated programs describe a valid Hoare logic proof if the rest of the proof tree's structure is implied
(caveats: Invariants are constrained, etc.)

(Informal) Rules for Decoration

Idea: check whether a decorated program represents a valid proof using local consistency checks

skip

pre and post-condition should be the same

$\{P\}$
skip
$\{P\}$

(skip) $\vdash \{P\}$ **skip** $\{P\}$

(Informal) Rules for Decoration

assignment

use the substitution from the rule

$$\{P[a/l]\}$$
$$l := a$$
$$\{P\}$$

(assign) $\vdash \{P[a/l]\}\ l := a\ \{P\}$

sequencing

$\{P\}\ c_1\ \{R\}$ and $\{R\}\ c_2\ \{Q\}$ should be (recursively) locally consistent

$$\{P\}$$
$$c_1\ ;$$
$$\{R\}$$
$$c_2$$
$$\{Q\}$$

(seq) $\dfrac{\vdash \{P\}\ c_1\ \{R\} \qquad \vdash \{R\}\ c_2\ \{Q\}}{\vdash \{P\}\ c_1\ ;\ c_2\ \{Q\}}$

(Informal) Rules for Decoration

if then

both branches are locally consistent; add condition to both

$\{P\}$
if b **then**
  $\{P \wedge b\}$
  c_1
  $\{Q\}$
else
  $\{P \wedge \neg b\}$
  c_2
  $\{Q\}$
$\{Q\}$

$$\text{(if)}\quad \frac{\vdash \{P \wedge b\}\ c_1\ \{Q\} \qquad \vdash \{P \wedge \neg b\}\ c_2\ \{Q\}}{\vdash \{P\}\ \textbf{if}\ b\ \textbf{then}\ c_1\ \textbf{else}\ c_2\ \{Q\}}$$

(Informal) Rules for Decoration

while
add/create loop invariant

$\{P\}$
while b **do**
  $\{P \land b\}$
  c
  $\{P\}$
$\{P \land \neg b\}$

(while) $$\frac{\vdash \{P \land b\}\ c\ \{P\}}{\vdash \{P\}\ \textbf{while}\ b\ \textbf{do}\ c\ \{P \land \neg b\}}$$

Australian National University

(Informal) Rules for Decoration

consequence

always write a (valid) implication

$$\{P\} \Rightarrow$$
$$\{P'\}$$

(cons) $$\frac{\models P \Rightarrow P' \qquad \vdash \{P'\}\, c\, \{Q'\} \qquad \models Q' \Rightarrow Q}{\vdash \{P\}\, c\, \{Q\}}$$

Floyd-Hoare Logic – Exercise

$$\{l_0 = n \wedge n > 0\}$$
$$l_1 := 1 \;;$$
while $!l_0 > 0$ **do**
$$l_1 := !l_1 \cdot l_0 \;;$$
$$l_0 := !l_0 - 1$$
$$\{l_1 = n!\}$$

Floyd-Hoare Logic – Exercise

$$\{l_0 = n \land n > 0\} \Rightarrow$$
$$\{1 = 1 \land l_0 = n \land n > 0\}$$
$$l_1 := 1\ ;$$
$$\{l_1 = 1 \land l_0 = n \land n > 0\} \Rightarrow$$
$$\{l_1 \cdot l_0! = n! \land l_0 \geq 0\}$$

while $!l_0 > 0$ **do**

$$\{l_1 \cdot l_0! = n! \land l_0 > 0 \land l_0 \geq 0\} \Rightarrow$$
$$\{l_1 \cdot l_0 \cdot (l_0 - 1)! = n! \land (l_0 - 1) \geq 0\}$$
$$l_1 := !l_1 \cdot l_0\ ;$$
$$\{l_1 \cdot (l_0 - 1)! = n! \land (l_0 - 1) \geq 0\}$$
$$l_0 := !l_0 - 1$$
$$\{l_1 \cdot l_0! = n! \land l_0 \geq 0\}$$

$$\{l_1 \cdot l_0! = n! \land (l_0 \geq 0) \land \neg(l_0 > 0)\} \Rightarrow$$
$$\{l_1 = n!\}$$