HOARE LOGIC

Assignment Rule example

$$\{\lambda(h,v).\ v(x)=0 \wedge v(y)=1\}$$
$$x \leftarrow y$$
$$\{\lambda(h,v).\ v(x)=1 \wedge v(y)=1\}$$

- Is this correct? not quite. Doesn't say anything about h.
  + they must be the same.
- should also generalize to arbitrary Precondition.

$$\{\overbrace{\lambda(h,v).\ v(x)=0 \wedge v(y)=1}^{P}\}$$
$$x \leftarrow y$$
$$\{\lambda(h,v).\ \exists v'.\ v=v'[x \mapsto [\![y]\!](h,v')] \wedge P\, h\, v'\}$$

we don't care about the details.

$$\{P\}$$
$$x \leftarrow e$$
$$\{\lambda(h,v).\ \exists v'.\ v=v'[x \mapsto [\![e]\!](h,v')] \wedge P\, h\, v'\}$$

while rule

while b do c

Consider b evaluates to false

$$\frac{}{\{P\}\ \text{while } b \text{ do } c\ \{\lambda s.\ \neg [\![b]\!](s) \land P(s)\}}$$

This is true when the loop terminates. But what about the body

$$\frac{\{\lambda s.\ P(s) \land [\![b]\!](s)\}\ c\ \{P\}}{\{P\}\ \text{while } b \text{ do } c\ \{\lambda s.\ \neg [\![b]\!](s) \land P(s)\}}$$

We will use a more general encoding

$$\frac{\forall s.\ P(s) \Rightarrow I(s) \quad \{\lambda s.\ I(s) \land [\![b]\!](s)\}\ c\ \{I\}}{\{P\}\ \{I\}\ \text{while } b \text{ do } c\ \{\lambda s.\ I(s) \land \neg [\![b]\!](s)\}}$$

"Induction hypothesis" for the loop.

+ Induction hypothesis must be Provided explicitly

Proving Programs using Hoare Logic

$$\mathrm{Swap} = tmp \leftarrow x;\quad x \leftarrow y;\quad y \leftarrow tmp$$

$$\{\lambda(h,v).\ v(x) = a \ \wedge\ v(y) = b\}$$

$$\mathrm{swap}$$

$$\{\lambda(h,v).\ v(x) = b \ \wedge\ v(y) = a\}$$

$$\{\overbrace{\lambda(h,v).\ v(x) = a \wedge v(y) = b}^{P}\}$$

$$tmp \leftarrow x$$

$$\{\lambda(h,v).\ \exists v_1.\ v = v_1[tmp \mapsto a] \ \wedge\ (* P\, h\, v_1 *)\ v_1(x) = a \ \wedge\ v_1(y) = b\}$$

$$x \leftarrow y$$

$$\{\lambda(h,v).\ \exists v_2.\ v = v_2[x \mapsto b] \ \wedge\ (* P\, h\, v_2 *)\ \exists v_1.\ v_2 = v_1[tmp \mapsto a] \ \wedge\ v_1(x) = a \ \wedge\ v_1(y) = b\}$$

$$y \leftarrow tmp$$

$$\{\lambda(h,v).\ \exists v_3.\ v = v_3[y \mapsto a] \ \wedge\ (P\, h\, v_3 *)\ \exists v_2.\ v_3 = v_2[x \to b] \ \wedge\ \exists v_1.\ v_2 = v_1[tmp \mapsto a] \ \wedge\ v_1(x) = a \ \wedge\ v_1(y) = b\}$$

Focussing on valuation only

$$v_1 = v_0[x \mapsto a][y \mapsto b]$$

$$v_2 = v_1[tmp \mapsto a] = v_0[x \mapsto a][y \mapsto b][tmp \mapsto a]$$

$$v_3 = v_2[x \mapsto b] = v_0[y \mapsto b][tmp \mapsto a][x \mapsto b]$$

$$v = v_3[y \mapsto a] = v_0[tmp \mapsto a][x \mapsto b][y \mapsto a]$$

$$///$$

$$\lambda(h,v).\ v(x) = b \ \wedge\ v(y) = a \ \wedge\ v(tmp) = a$$

$\lambda(h,v).\ v(x) = b \wedge v(y$

$\rightarrow \lambda(h,v).\ v(x) = b \wedge v(y) = a$ (* original Post condition *)

$\hookrightarrow$ necessity for rule of consequence.