Hoare logic

$\{P\}\ S\ \{Q\}$ → encode in logic

logic FOL

Simple programming language (no loops)

$P ::= x := a$

variable

arithmetic expression

$\mid$ if b then P_1 else P_2

$\mid P_1 ; P_2$

Recall operational semantics

$\langle P, s\rangle \longrightarrow s'$

V is the set of all program variables

state $s : V \mapsto \mathbb{Z}$

Transition relation

$T \subseteq State \times State$

set of all states

$T(V, V')$ $\qquad T(s, s')$

E.g. $\quad x := x + 1$

$T \subseteq \mathbb{Z} \times \mathbb{Z}$

$T = \{ (n, n+1) \mid n \in \mathbb{Z} \}$

x	x'
0	1
1	2
2	3
$\vdots$	$\vdots$

$V = \{x\}$ $V' = \{x'\}$

$T(x, x')$

FOL / theory linear integer arithmetic (LIA)

$$\begin{aligned}
\varphi ::= \; & a_1 = a_2 \\
\mid \; & a_1 \leq a_2 \\
\mid \; & \varphi_1 \wedge \varphi_2 \\
\mid \; & \varphi_1 \vee \varphi_2 \\
\mid \; & \neg \varphi \\
\mid \; & \exists x . \varphi \\
\mid \; & \forall x . \varphi
\end{aligned}$$

integers

$c_1 x_1 + c_2 x_2 + \cdots + c x_n$

variables

E.g. $x + y > 0$

$m = \{ x \mapsto 0, y \mapsto 1 \}$

$m \models x + y > 0$

Encode the transition relation

① $x := a$

E.g. $x := x+1$

x	x'
0	1
2	3
15	16
⋮	⋮

$x' = x+1$

$m \models x' = x+1$

$m = \{x' \mapsto 1,\ x \mapsto 0\}$

Encoding $x := a$

$$enc(x := a) \triangleq x' = a \wedge \underbrace{\bigwedge_{y \in V - \{x\}} y' = y}_{\text{frame axiom}}$$

E.g.

$x := x + y$

$V = \{x, y\}$

x	y	x'	y'
0	0	0	0
0	1	1	1
⋮	⋮	⋮	⋮

$T(x, y, x', y') \triangleq x' = x + y \wedge y' = y$

$T(V, V')$

Encode if statement

if $\underset{x>0}{b}$ then P_1 else P_2

$$enc(\text{if} \ldots) \triangleq (b \Rightarrow enc(P_1)) \wedge (\neg b \Rightarrow enc(P_2))$$

$$\equiv (b \wedge enc(P_1)) \vee (\neg b \wedge enc(P_2))$$

E.g. if $x > 0$ then $x = x+1$ else $x = y$

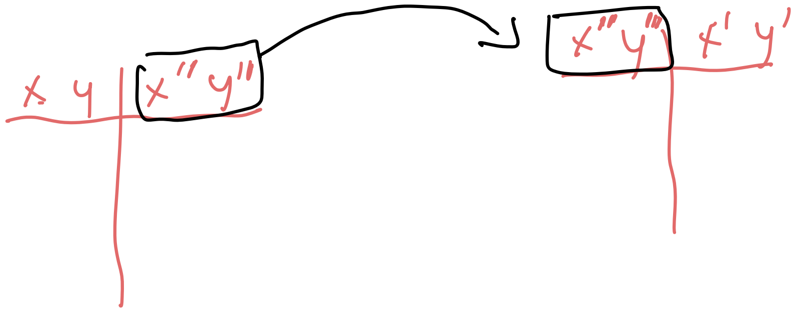
$$enc(\text{if} \ldots) \triangleq (x > 0 \Rightarrow (x' = x+1 \wedge y' = y))$$

$$\wedge (x \leq 0 \Rightarrow (x' = y \wedge y' = y))$$

$P_1 \; ; \; P_2$ $\qquad$ $enc : \text{Program} \longrightarrow \text{FOL}$

E.g. $\quad x = x+1 \; ; \; y = y+1$

$$x = x+1 \xrightarrow{enc} \qquad y = y+1 \xrightarrow{enc}$$



$$\exists x'', y''. \begin{pmatrix} x'' = x+1 \wedge \\ y'' = y \end{pmatrix} \wedge \begin{pmatrix} y' = y''+1 \wedge \\ x' = x'' \end{pmatrix}$$

$T(V, V')$

---

general form

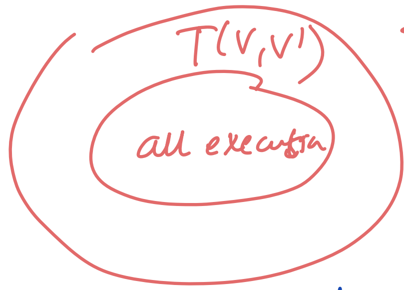
$$enc(P_1 \; ; \; P_2) \triangleq$$

$$\exists V''. \; T_1(V, V'') \wedge T_2(V'', V')$$

where $T_1(V, V') = enc(P_1)$

$T_2(V, V') = enc(P_2)$

Soundness / Completeness



sound but incomplete

$s : V \mapsto \mathbb{Z}$

__Completeness:__ Fix a program P

let $m \models enc(P)$

$s = \{ v \mapsto m(v) \mid v \in V \}$

$s' = \{ v' \mapsto m(v') \mid v \in V \}$

Then, $\langle P, s \rangle \to s'$

__soundness:__ Let $\langle P, s \rangle \to s'$

let $m = \{ v \mapsto s(v) \mid v \in V \} \cup$

$\{ v' \mapsto s'(v) \mid v \in V \}$

Then $m \models enc(P)$

__Verification__

$\{ \phi \} \; P \; \{ \psi \}$

LIA   LIA

e.g. $x > 0$

for any state $s \in \phi$, if $\langle P, s \rangle \to s'$, then $s' \in \psi$

$$\phi \wedge enc(P) \Rightarrow \psi' \quad \text{is VALID}$$

iff

$$\{\phi\}\ P\ \{\psi\} \quad \text{is VALID/holds}$$

---

e.g. $\{x>0\}\ x:=x+1\ \{x>1\}$

$x>0 \wedge x'=x+1 \Rightarrow x'>1$ is VALID ✓

$\{x>0\}\ x:=x+y\ \{x>1\}$

$$x>0 \wedge \underbrace{x'=x+y \wedge y'=y}_{enc} \Rightarrow x'>1$$

$x \mapsto 1$

$y \mapsto 0$

$y' \mapsto 0$

$x' \mapsto 1$

Bounded model checking (BMC)

Symbolic execution (SE)