# Verification + Hoare Logic

Verification - prove facts about a program
(ex. proved that well-typed programs don't get stuck)
Other forms of verification can prove more complex, interesting things about programs

Hoare Logic
(Sir Charles Antony Richard Hoare
(Also known for Quicksort, Null pointer, ALGOL, dining philosophers)
Prove logical assertions about programs

Programs - Remember IMP

$E ::= \bar{n} \mid E+E$
$B ::= \text{true} \mid \text{false} \mid E=E$
$S ::= \text{skip} \mid x := E \mid \text{if } B \text{ then } S \text{ else } S \mid \text{while } B \text{ do } S \mid S;S$

Logical Assertions
Predicate logic

$P ::= A \mid P \wedge P \mid P \vee P \mid \neg P \mid P \rightarrow P \mid P \leftrightarrow P$
Atomic props. and or not implies "if and only if"

$\mid \forall x. P \mid \exists x. P \mid P(x)$
for all there exists

"It is Tuesday" $\wedge$ "This is CS440"
("It is Tuesday" $\wedge$ "It is 10:00-11:15" $\wedge$ "In SB 104") $\rightarrow$ "CS440"

Quantifiers

$\forall x \in \mathbb{Z}. \exists y \in \mathbb{Z}. y > x$ — "for any integer $x$, there is an integer $y$ s.t. $y > x$"

Make assertions about stores $\sigma$

$\{x \mapsto 0\}$ true assertions: $x = 0$, $x \geq 0$, $x > -5$, ...
false " : $x > 0$, $x$ is odd, ...

$\sigma \models P$ "$\sigma$ satisfies $P$" - $P$ is true in $\sigma$
$\models P$ "$P$ is valid" - holds in any store

$\{x \mapsto 0\} \models x \geq 0$ $\{x \mapsto 5, y \mapsto 0\} \models x \geq 0$
$\models x > 0 \rightarrow x \geq 0$ $\models$ true

Making assertions about programs
Hoare triple $\{P\}\ S\ \{Q\}$
preconditions, program, postconditions

"If $P$ holds before running $S$ and $S$ terminates, then $Q$ holds after running $S$."
"partial correctness"

$\sigma \models \{P\}\ S\ \{Q\}$ - triple holds under $\sigma$
If $\sigma \models P$ and $\langle \sigma, S \rangle \Downarrow \sigma'$ then $\sigma' \models Q$

$\models \{P\}\ S\ \{Q\}$ - triple holds for any $\sigma$ that satisfies $P$
$\forall \sigma$, if $\sigma \models P$ and $\langle \sigma, S \rangle \Downarrow \sigma'$ then $\sigma' \models Q$

e.g. $\{x=0\}\ x:=x+1\ \{x=1\}$ ✓
$\{x=0\}\ x:=x+1\ \{x<0\}$ ✗
$\{x=0\}\ x:=x+1\ \{x>0\}$ ✓ (but weaker)
$\{x>0\}\ x:=x+1\ \{x>0\}$ ✓
$\{x\geq 0\}\ x:=x+1\ \{x>0\}$ ✓

$\{true\}\ x:=y/z\ \{z*x=y\}$ ?
integer div

$z=0$ ? Ok, b.c. then S doesn't terminate
$\langle\{y=3, z=2\}, S\rangle \Downarrow \{y=3, z=2, x=1\}$ ✗

Options: 1. Make precond. stronger (more restrictive)
$\{y=kz\}\ x:=y/z\ \{x=k\}$
"ghost variable"

2. Make postcond. weaker
$\{true\}\ x:=y/z\ \{z*x\leq y<z*(x+1)\}$

3. Fix the program
$\{true\}$ ???