HARVARD
John A. Paulson
School of Engineering
and Applied Sciences

Some background info for program synthesis

CS 252, Fall 2017

Topics

- Hoare logic
  - Reasoning about programs
  - Weakest precondition
  - See also lecture notes for CS152 in Spring 2014
    - https://www.seas.harvard.edu/courses/cs152/2014sp/
- Abstract interpretation
  - Approximating concrete execution
  - See lecture notes for CS252 in Spring 2011
    - https://www.seas.harvard.edu/courses/cs252/2011sp/
  - See also lecture notes for CS152 in Spring 2014
    - https://www.seas.harvard.edu/courses/cs152/2014sp/
- Model checking
  - See lecture notes for CS252 in Spring 2011
    - https://www.seas.harvard.edu/courses/cs252/2011sp/

Axiomatic Semantics

- Key idea: give specifications for what programs are supposed to do
  - Define meaning of programs in terms of logical formulas satisfied by program
  - Enables reasoning about programs
- Pre- and post-condition:

$$\{Pre\}\; c\; \{Post\}$$

  - Partial correctness: "If *Pre* holds before execution of *c*, and *c* terminates, then *Post* holds after *c*."
  - (Total correctness: "If *Pre* holds before execution of *c* then c terminates and *Post* holds after *c*.")

Example

- Example

$$\{\mathsf{foo} = 0 \land \mathsf{bar} = i\}\ \mathsf{baz} := 0; \textbf{while}\ \mathsf{foo} \neq \mathsf{bar}\ \textbf{do}\ (\mathsf{baz} := \mathsf{baz} - 2; \mathsf{foo} := \mathsf{foo} + 1)\ \{\mathsf{baz} = -2i\}$$

- Non example
  - { true } if foo < 0 then foo := -foo else skip { foo > 0 }

Hoare Logic Rules

$$\textsc{Skip}\ \frac{}{\vdash \{P\}\ \textbf{skip}\ \{P\}}$$

$$\textsc{Assign}\ \frac{}{\vdash \{P[a/x]\}\ x := a\ \{P\}}$$

$$\textsc{Seq}\ \frac{\vdash \{P\}\ c_1\ \{R\} \qquad \vdash \{R\}\ c_2\ \{Q\}}{\vdash \{P\}\ c_1; c_2\ \{Q\}}$$

$$\textsc{If}\ \frac{\vdash \{P \land b\}\ c_1\ \{Q\} \qquad \vdash \{P \land \neg b\}\ c_2\ \{Q\}}{\vdash \{P\}\ \textbf{if}\ b\ \textbf{then}\ c_1\ \textbf{else}\ c_2\ \{Q\}}$$

$$\textsc{While}\ \frac{\vdash \{P \land b\}\ c\ \{P\}}{\vdash \{P\}\ \textbf{while}\ b\ \textbf{do}\ c\ \{P \land \neg b\}}$$

$$\textsc{Consequence}\ \frac{\vDash (P \Rightarrow P') \qquad \vdash \{P'\}\ c\ \{Q'\} \qquad \vDash (Q' \Rightarrow Q)}{\vdash \{P\}\ c\ \{Q\}}$$

- Hoare logic is sound and **relatively complete**
  - No more incomplete that our language of assertions $\vDash P \Rightarrow Q$

Hoare Logic Rules

$$\textsc{Skip}\ \frac{}{\vdash \{P\}\ \textbf{skip}\ \{P\}}$$

$$\textsc{Assign}\ \frac{}{\vdash \{P[a/x]\}\ x := a\ \{P\}}$$

$$\textsc{Seq}\ \frac{\vdash \{P\}\ c_1\ \{R\} \qquad \vdash \{R\}\ c_2\ \{Q\}}{\vdash \{P\}\ c_1; c_2\ \{Q\}}$$

$$\textsc{If}\ \frac{\vdash \{P \land b\}\ c_1\ \{Q\} \qquad \vdash \{P \land \neg b\}\ c_2\ \{Q\}}{\vdash \{P\}\ \textbf{if}\ b\ \textbf{then}\ c_1\ \textbf{else}\ c_2\ \{Q\}}$$

$$\textsc{While}\ \frac{\vdash \{P \land b\}\ c\ \{P\}}{\vdash \{P\}\ \textbf{while}\ b\ \textbf{do}\ c\ \{P \land \neg b\}}$$

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

$$\{\mathsf{foo} = 0 \land \mathsf{bar} = i\}\ \mathsf{baz} := 0; \textbf{while}\ \mathsf{foo} \neq \mathsf{bar}\ \textbf{do}\ (\mathsf{baz} := \mathsf{baz} - 2; \mathsf{foo} := \mathsf{foo} + 1)\ \{\mathsf{baz} = -2i\}$$

Example

- Build a proof tree for the following:

$$\{\mathsf{foo} = 0 \wedge \mathsf{bar} = i\}\ \mathsf{baz} := 0; \textbf{while}\ \mathsf{foo} \neq \mathsf{bar}\ \textbf{do}\ (\mathsf{baz} := \mathsf{baz} - 2; \mathsf{foo} := \mathsf{foo} + 1)\ \{\mathsf{baz} = -2i\}$$

Predicate transformation

- We now have a logic to prove partial correctness triples {P} *c* {Q}
- Interesting question: Given Q and *c*, what is the weakest P such that {P} *c* {Q} ?
  - ***Weakest (liberal) pre-condition***
  - E.g., Consider *c*≡ “a = int[50]; i =0; while (i < b) { ... }; a[i]=0”
  - What is the weakest precondition P such that {P} c { i ≥ 50 }? i.e., how do we trigger an overflow?
- Dual is ***strongest post-condition***: given P and *c*, what is the strongest Q such that {P} *c* {Q} ?

Weakest pre-condition

- wp(c, Q) = P where P is the weakest condition such that {P} c {Q}
- wp(skip, Q) = Q
- wp(x := e, Q) = Q{e/x}
  - e.g., wp(foo := bar+1, foo > 42) = (bar+1 > 42)
- wp(c1;c2, Q) = wp(c1, wp(c2, Q))
- wp(if b then c1 else c2,Q) = b⇒wp(c1, Q) ∧ ¬b⇒wp(c2, Q)
  - e.g., wp(if x < 0 then x := -x else skip, x > 0)=?

Weakest pre-condition

- wp(while b do c, Q) = ???
- In general undecidable
- Conservative under approximation: unroll loop
  - wp′(while b do c, Q) =

    wp(if (b) then (c;if(b) then c), Q∧¬b)
    - i.e., approximate 0-2 executions of loop
  - {P} while b do c {Q} is valid if

    P⇒wp′(while b do c, Q)
    - The converse if not necessarily true

Weakest pre-condition

- wp(while b do c, Q) = ???
- Conservative under approximation: loop invariant
  - A loop invariant *I* is true at top of each loop iteration
  - Loop invariant typically supplied by programmer, or use heuristics to guess
  - wp′(while b do c, Q) =

	$I \wedge b \Rightarrow wp(c, I)$	*I is a loop invariant*
$\wedge$	$(\neg b \wedge Q \vee$	*loop won′t execute*
	$(I \wedge (I \wedge \neg b \Rightarrow Q))$	*Invariant holds*
		and Q holds when loop exits

- *Note this is **weakest liberal precondition**: it does not require termination*