NICTA

COMP 4161

NICTA Advanced Course

Advanced Topics in Software Verification

Gerwin Klein, June Andronick, Toby Murray, Rafal Kolanski

$$\{P\} \ldots \{Q\}$$

Last Time

- Calculations: also/finally
- [trans]-rules
- Code generation

Content

- ➜ Intro & motivation, getting started [1]

- ➜ Foundations & Principles
  - Lambda Calculus, natural deduction [1,2]
  - Higher Order Logic [3^a]
  - Term rewriting [4]

- ➜ Proof & Specification Techniques
  - Isar [5]
  - Inductively defined sets, rule induction [6^b]
  - Datatypes, recursion, induction [7^c, 8]
  - Calculational reasoning, code generation [9]
  - Hoare logic, proofs about programs [10^d,11,12]

^{a}a1 due; ^{b}a2 due; ^{c}session break; ^{d}a3 due

NICTA

A Crash Course in Semantics

IMP - a small Imperative Language

Commands:

datatype com	=	SKIP	
	|	Assign loc aexp	$(\_ := \_)$
	|	Semi com com	$(\_; \_)$
	|	Cond bexp com com	(IF _ THEN _ ELSE _)
	|	While bexp com	(WHILE _ DO _ OD)

types loc	=	string
types state	=	loc $\Rightarrow$ nat

types aexp	=	state $\Rightarrow$ nat
types bexp	=	state $\Rightarrow$ bool

Example Program

NICTA

Usual syntax:

$$
\begin{array}{l}
B := 1; \\
\textsf{WHILE } A \neq 0 \textsf{ DO} \\
\quad B := B * A; \\
\quad A := A - 1 \\
\textsf{OD}
\end{array}
$$

Expressions are functions from state to bool or nat:

$$
\begin{array}{l}
B := (\lambda\sigma.\ 1); \\
\textsf{WHILE } (\lambda\sigma.\ \sigma\ A \neq 0) \textsf{ DO} \\
\quad B := (\lambda\sigma.\ \sigma\ B * \sigma\ A); \\
\quad A := (\lambda\sigma.\ \sigma\ A - 1) \\
\textsf{OD}
\end{array}
$$

What does it do?

So far we have defined:

- ➜ **Syntax** of commands and expressions
- ➜ **State** of programs (function from variables to values)

Now we need: the meaning (semantics) of programs

How to define execution of a program?

- ➜ A wide field of its own
- ➜ Some choices:
  - Operational (inductive relations, big step, small step)
  - Denotational (programs as functions on states, state transformers)
  - Axiomatic (pre-/post conditions, Hoare logic)

Structural Operational Semantics

NICTA

$$\frac{}{\langle \mathsf{SKIP}, \sigma \rangle \longrightarrow \sigma}$$

$$\frac{e\ \sigma = v}{\langle \mathsf{x} := \mathsf{e}, \sigma \rangle \longrightarrow \sigma[x \mapsto v]}$$

$$\frac{\langle c_1, \sigma \rangle \longrightarrow \sigma' \quad \langle c_2, \sigma' \rangle \longrightarrow \sigma''}{\langle c_1; c_2, \sigma \rangle \longrightarrow \sigma''}$$

$$\frac{b\ \sigma = \mathsf{True} \quad \langle c_1, \sigma \rangle \longrightarrow \sigma'}{\langle \mathsf{IF}\ b\ \mathsf{THEN}\ c_1\ \mathsf{ELSE}\ c_2, \sigma \rangle \longrightarrow \sigma'}$$

$$\frac{b\ \sigma = \mathsf{False} \quad \langle c_2, \sigma \rangle \longrightarrow \sigma'}{\langle \mathsf{IF}\ b\ \mathsf{THEN}\ c_1\ \mathsf{ELSE}\ c_2, \sigma \rangle \longrightarrow \sigma'}$$

Structural Operational Semantics

NICTA

$$\frac{b\ \sigma = \text{False}}{\langle \text{WHILE } b \text{ DO } c \text{ OD}, \sigma \rangle \longrightarrow \sigma}$$

$$\frac{b\ \sigma = \text{True} \quad \langle c, \sigma \rangle \longrightarrow \sigma' \quad \langle \text{WHILE } b \text{ DO } c \text{ OD}, \sigma' \rangle \longrightarrow \sigma''}{\langle \text{WHILE } b \text{ DO } c \text{ OD}, \sigma \rangle \longrightarrow \sigma''}$$

Demo: The Definitions in Isabelle

Proofs about Programs

NICTA

Now we know:

- ➜ What programs are: Syntax
- ➜ On what they work: State
- ➜ How they work: Semantics

So we can prove properties about programs

Example:

Show that example program from slide 6 implements the factorial.

$$\textbf{lemma}\ \langle \mathsf{factorial}, \sigma \rangle \longrightarrow \sigma' \Longrightarrow \sigma' B = \mathsf{fac}\ (\sigma A)$$

$$(\text{where} \quad \mathsf{fac}\ 0 = 1, \quad \mathsf{fac}\ (\mathsf{Suc}\ n) = (\mathsf{Suc}\ n) * \mathsf{fac}\ n)$$

Demo: Example Proof

Too tedious

Induction needed for each loop

Is there something easier?

Floyd/Hoare

NICTA

Idea: describe meaning of program by pre/post conditions

Examples:

$\{\mathsf{True}\} \quad x := 2 \quad \{x = 2\}$

$\{y = 2\} \quad x := 21 * y \quad \{x = 42\}$

$\{x = n\} \quad \mathsf{IF}\ y < 0\ \mathsf{THEN}\ x := x + y\ \mathsf{ELSE}\ x := x - y \quad \{x = n - |y|\}$

$\{A = n\} \quad \mathsf{factorial} \quad \{B = \mathsf{fac}\ n\}$

Proofs: have rules that directly work on such triples

Meaning of a Hoare-Triple

$$\{P\} \quad c \quad \{Q\}$$

What are the assertions P and Q?

- ➜ Here: again functions from state to bool
  (shallow embedding of assertions)
- ➜ Other choice: syntax and semantics for assertions (deep embedding)

What does $\{P\}\ c\ \{Q\}$ mean?

Partial Correctness:

$$\models \{P\}\ c\ \{Q\} \quad \equiv \quad (\forall \sigma\ \sigma'.\ P\ \sigma \wedge \langle c, \sigma\rangle \longrightarrow \sigma' \Longrightarrow Q\ \sigma')$$

Total Correctness:

$$\models \{P\}\ c\ \{Q\} \quad \equiv \quad (\forall \sigma.\ P\ \sigma \Longrightarrow \exists \sigma'.\ \langle c, \sigma\rangle \longrightarrow \sigma' \wedge Q\ \sigma')$$

This lecture: partial correctness only (easier)

Hoare Rules

$$\frac{}{\{P\} \quad \mathsf{SKIP} \quad \{P\}} \qquad \frac{}{\{P[x \mapsto e]\} \quad x := e \quad \{P\}}$$

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

$$\frac{\{P \wedge b\}\ c_1\ \{Q\} \quad \{P \wedge \neg b\}\ c_2\ \{Q\}}{\{P\} \quad \mathsf{IF}\ b\ \mathsf{THEN}\ c_1\ \mathsf{ELSE}\ c_2 \quad \{Q\}}$$

$$\frac{\{P \wedge b\}\ c\ \{P\} \quad P \wedge \neg b \Longrightarrow Q}{\{P\} \quad \mathsf{WHILE}\ b\ \mathsf{DO}\ c\ \mathsf{OD} \quad \{Q\}}$$

$$\frac{P \Longrightarrow P' \quad \{P'\}\ c\ \{Q'\} \quad Q' \Longrightarrow Q}{\{P\} \quad c \quad \{Q\}}$$

Hoare Rules

$$\frac{}{\vdash \{P\} \quad \mathsf{SKIP} \quad \{P\}} \qquad \frac{}{\vdash \{\lambda\sigma.\ P\ (\sigma(x := e\ \sigma))\} \quad x := e \quad \{P\}}$$

$$\frac{\vdash \{P\}\ c_1\ \{R\} \quad \vdash \{R\}\ c_2\ \{Q\}}{\vdash \{P\} \quad c_1; c_2 \quad \{Q\}}$$

$$\frac{\vdash \{\lambda\sigma.\ P\ \sigma \wedge b\ \sigma\}\ c_1\ \{R\} \quad \vdash \{\lambda\sigma.\ P\ \sigma \wedge \neg b\ \sigma\}\ c_2\ \{Q\}}{\vdash \{P\} \quad \mathsf{IF}\ b\ \mathsf{THEN}\ c_1\ \mathsf{ELSE}\ c_2 \quad \{Q\}}$$

$$\frac{\vdash \{\lambda\sigma.\ P\ \sigma \wedge b\ \sigma\}\ c\ \{P\} \quad \bigwedge\sigma.\ P\ \sigma \wedge \neg b\ \sigma \Longrightarrow Q\ \sigma}{\vdash \{P\} \quad \mathsf{WHILE}\ b\ \mathsf{DO}\ c\ \mathsf{OD} \quad \{Q\}}$$

$$\frac{\bigwedge\sigma.\ P\ \sigma \Longrightarrow P'\ \sigma \quad \vdash \{P'\}\ c\ \{Q'\} \quad \bigwedge\sigma.\ Q'\ \sigma \Longrightarrow Q\ \sigma}{\vdash \{P\} \quad c \quad \{Q\}}$$

Are the Rules Correct?

Soundness: $\vdash \{P\}\ c\ \{Q\} \Longrightarrow \models \{P\}\ c\ \{Q\}$

Proof: by rule induction on $\vdash \{P\}\ c\ \{Q\}$

Demo: Hoare Logic in Isabelle