COMP 4161
NICTA Advanced Course

Advanced Topics in Software Verification

Gerwin Klein, June Andronick, Toby Murray

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

Slide 1

Last Time

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

Slide 2

Content

	Rough timeline
➜ Intro & motivation, getting started	[1]
➜ Foundations & Principles	
• Lambda Calculus, natural deduction	[2,3,4[a]]
• Higher Order Logic	[5,6[b],7]
• Term rewriting	[8,9,10[c]]
➜ Proof & Specification Techniques	
• Isar	[11,12[d]]
• Inductively defined sets, rule induction	[13[e],15]
• Datatypes, recursion, induction	[16,17[f],18,19]
• Calculational reasoning, mathematics style proofs	[20]
• Hoare logic, proofs about programs	[21[g],22,23]

[a] a1 out; [b] a1 due; [c] a2 out; [d] a2 due; [e] session break; [f] a3 out; [g] a3 due

Slide 3

A Crash Course in Semantics

Slide 4

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 ⇒ nat	
types aexp	=	state ⇒ nat	
types bexp	=	state ⇒ bool	

Example Program

Usual syntax:

$B := 1;$
WHILE $A \neq 0$ DO
  $B := B * A;$
  $A := A - 1$
OD

Expressions are functions from state to bool or nat:

$B := (\lambda\sigma.\ 1);$
WHILE $(\lambda\sigma.\ \sigma\ A \neq 0)$ DO
  $B := (\lambda\sigma.\ \sigma\ B * \sigma\ A);$
  $A := (\lambda\sigma.\ \sigma\ A - 1)$
OD

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

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

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

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

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

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

Structural Operational Semantics

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

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

Slide 9

Demo: The Definitions in Isabelle

Slide 10

Proofs about Programs

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.

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

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

Slide 11

Demo: Example Proof

Slide 12

Too tedious

Induction needed for each loop

Is there something easier?

Slide 13

Floyd/Hoare

Idea: describe meaning of program by pre/post conditions

Examples:

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

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

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

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

Proofs: have rules that directly work on such triples

Slide 14

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)

Slide 15

Hoare Rules

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

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

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

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

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

Slide 16

Hoare Rules

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

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

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

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

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

Slide 17

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

Slide 18