THE UNIVERSITY *of* EDINBURGH
informatics

cisa | Centre for Intelligent Systems and their Applications

Program verification using Hoare Logic[1]

Automated Reasoning - Guest Lecture

Petros Papapanagiotou
pe.p@ed.ac.uk

Part 2 of 2

[1] Contains material from Mike Gordon's slides: http://www.cl.cam.ac.uk/~mjcg/HL

Previously on Hoare Logic

A simple "while" language

- Sequence: `a ; b`
- Skip (do nothing): `SKIP`
- Variable assignment: `X := 0`
- Conditional:
  `IF cond THEN a ELSE b FI`
- Loop: `WHILE cond DO c OD`

Hoare Logic

- $\{P\}\ C\ \{Q\}$
- Formal specification
- Axiomatic semantics
- Hoare Logic Rules and examples

Conditional Rule

$$\frac{\{P \wedge S\}\ C_1\ \{Q\} \quad \{P \wedge \neg S\}\ C_2\ \{Q\}}{\{P\}\ \texttt{IF}\ S\ \texttt{THEN}\ C_1\ \texttt{ELSE}\ C_2\ \texttt{FI}\ \{Q\}}$$

- Example (`Max X Y`):

$$\frac{}{\{X \geq X \wedge X \geq Y\}\ \texttt{MAX} := \texttt{X}\ \{MAX \geq X \wedge MAX \geq Y\}}$$
$$\{\mathbf{T} \wedge X \geq Y\}\ \texttt{MAX} := \texttt{X}\ \{MAX \geq X \wedge MAX \geq Y\} \tag{1}$$

$$\frac{}{\{Y \geq X \wedge Y \geq Y\}\ \texttt{MAX} := \texttt{Y}\ \{MAX \geq X \wedge MAX \geq Y\}}$$
$$\{\mathbf{T} \wedge \neg(X \geq Y)\}\ \texttt{MAX} := \texttt{Y}\ \{MAX \geq X \wedge MAX \geq Y\} \tag{2}$$

$$\frac{(1) \qquad (2)}{\{\mathbf{T}\}\ \texttt{IF X} \geq \texttt{Y THEN MAX} := \texttt{X ELSE MAX} := \texttt{Y FI}\ \{MAX \geq X \wedge MAX \geq Y\}} \tag{3}$$

What if?

$$\frac{\{P \wedge S\}\ C_1\ \{Q\} \quad \{P \wedge \neg S\}\ C_2\ \{Q\}}{\{P\}\ \texttt{IF}\ S\ \texttt{THEN}\ C_1\ \texttt{ELSE}\ C_2\ \texttt{FI}\ \{Q\}}$$

- Example (`Max X Y`):

$$\frac{\dfrac{}{\{X = max(X, Y)\}\ \texttt{MAX} := \texttt{X}\ \{MAX = max(X, Y)\}}}{\dfrac{???}{\{\mathbf{T} \wedge X \geq Y\}\ \texttt{MAX} := \texttt{X}\ \{MAX = max(X, Y)\}}} \tag{4}$$

$$\frac{\dfrac{}{\{Y = max(X, Y)\}\ \texttt{MAX} := \texttt{Y}\ \{MAX = max(X, Y)\}}}{\dfrac{???}{\{\mathbf{T} \wedge \neg(X \geq Y)\}\ \texttt{MAX} := \texttt{Y}\ \{MAX = max(X, Y)\}}} \tag{5}$$

$$\frac{(4) \qquad (5)}{\{\mathbf{T}\}\ \texttt{IF X} \geq \texttt{Y THEN MAX} := \texttt{X ELSE MAX} := \texttt{Y FI}\ \{MAX = max(X, Y)\}} \tag{6}$$

Precondition Strengthening

$$\frac{P \longrightarrow P' \quad \{P'\}\ C\ \{Q\}}{\{P\}\ C\ \{Q\}}$$

- Replace a *precondition* with a stronger condition
- Example:

$$\frac{X = n \longrightarrow X + 1 = n + 1 \quad \dfrac{}{\{X + 1 = n + 1\}\ \texttt{X} := \texttt{X} + \texttt{1}\ \{X = n + 1\}}}{\{X = n\}\ \texttt{X} := \texttt{X} + \texttt{1}\ \{X = n + 1\}}$$

Postcondition Weakening

$$\frac{\{P\}\ C\ \{Q'\} \quad Q' \longrightarrow Q}{\{P\}\ C\ \{Q\}}$$

- Replace a *postcondition* with a weaker condition
- Example:

$$\frac{\{X = n\}\ \texttt{X} := \texttt{X} + \texttt{1}\ \{X = n + 1\} \quad X = n + 1 \longrightarrow X > n}{\{X = n\}\ \texttt{X} := \texttt{X} + \texttt{1}\ \{X > n\}}$$

Aha!

$$\frac{P \longrightarrow P' \quad \{P'\}\ C\ \{Q\}}{\{P\}\ C\ \{Q\}}$$

- Example (`Max X Y`):

$$\frac{\mathbf{T} \wedge X \geq Y \longrightarrow X = max(X, Y) \quad \overline{\{X = max(X, Y)\}\ \texttt{MAX} := \texttt{X}\ \{MAX = max(X, Y)\}}}{\{\mathbf{T} \wedge X \geq Y\}\ \texttt{MAX} := \texttt{X}\ \{MAX = max(X, Y)\}} \tag{7}$$

$$\frac{\mathbf{T} \wedge \neg(X \geq Y) \longrightarrow Y = max(X, Y) \quad \overline{\{Y = max(X, Y)\}\ \texttt{MAX} := \texttt{Y}\ \{MAX = max(X, Y)\}}}{\{\mathbf{T} \wedge \neg(X \geq Y)\}\ \texttt{MAX} := \texttt{Y}\ \{MAX = max(X, Y)\}} \tag{8}$$

$$\frac{(7) \qquad (8)}{\{\mathbf{T}\}\ \texttt{IF X} \geq \texttt{Y THEN MAX} := \texttt{X ELSE MAX} := \texttt{Y FI}\ \{MAX = max(X, Y)\}} \tag{9}$$

Verification Conditions (VCs)

$$\{\mathbf{T}\}\ \texttt{IF X} \geq \texttt{Y THEN MAX} := \texttt{X ELSE MAX} := \texttt{Y FI}\ \{MAX = max(X, Y)\}$$

- FOL VCs:

$$\mathbf{T} \wedge X \geq Y \longrightarrow X = max(X, Y)$$

$$\mathbf{T} \wedge \neg(X \geq Y) \longrightarrow Y = max(X, Y)$$

- Hoare Logic rules can be applied *automatically* to generate VCs
  - e.g. Isabelle's `vcg` tactic
- We need to provide proofs for the VCs / *proof obligations*
  - Reduced to FOL statements
  - From simple algebraic proofs to reasoning about inductive data types

WHILE Rule

$$\frac{\{P \wedge S\}\ C\ \{P\}}{\{P\}\ \texttt{WHILE}\ S\ \texttt{DO}\ C\ \texttt{OD}\ \{P \wedge \neg S\}}$$

- P is an *invariant* for C whenever S holds
- *WHILE rule*: If executing C *once* preserves the truth of P, then executing C *any number of times* also preserves the truth of P
- If P is an invariant for C when S holds then P is an invariant of the *whole* WHILE loop, i.e. a *loop invariant*

WHILE Rule

$$\frac{\{P \wedge S\}\ C\ \{P\}}{\{P\}\ \texttt{WHILE}\ S\ \texttt{DO}\ C\ \texttt{OD}\ \{P \wedge \neg S\}}$$

$\{\mathbf{Y = 1 \wedge Z = 0}\}$
`WHILE Z ≠ X DO`
`  Z := Z + 1 ;`
`  Y := Y × Z`
`OD`
$\{\mathbf{Y = X!}\}$

vs.

$\{P\}$
`WHILE Z ≠ X DO`
`  Z := Z + 1 ;`
`  Y := Y × Z`
`OD`
$\{P \wedge \neg Z \neq X\}$

- What is P?

WHILE Rule - How to find an invariant

$$\frac{\{P \wedge S\}\ C\ \{P\}}{\{P\}\ \texttt{WHILE}\ S\ \texttt{DO}\ C\ \texttt{OD}\ \{P \wedge \neg S\}}$$

- The invariant P should:
  - Say what *has been done so far* together with what *remains to be done*
  - Hold *at each iteration* of the loop.
  - Give the *desired result* when the loop terminates

WHILE Rule - Invariant VCs

$$\frac{\{P \wedge S\}\ C\ \{P\}}{\{P\}\ \texttt{WHILE}\ S\ \texttt{DO}\ C\ \texttt{OD}\ \{P \wedge \neg S\}}$$

$$\{\mathbf{Y} = \mathbf{1} \wedge \mathbf{Z} = \mathbf{0}\}\ \texttt{WHILE Z} \neq \texttt{X DO Z} := \texttt{Z} + \mathbf{1}\ ;\ \texttt{Y} := \texttt{Y} \times \texttt{Z OD}\ \{\mathbf{Y} = \mathbf{X}!\}$$

$$\{P\}\ \texttt{WHILE Z} \neq \texttt{X DO Z} := \texttt{Z} + \mathbf{1}\ ;\ \texttt{Y} := \texttt{Y} \times \texttt{Z OD}\ \{P \wedge \neg Z \neq X\}$$

- We need to find an invariant P such that:
  - $\{P \wedge Z \neq X\}\ \texttt{Z} := \texttt{Z} + \mathbf{1}\ ;\ \texttt{Y} := \texttt{Y} \times \texttt{Z}\ \{P\}$ (WHILE rule)
  - $\mathbf{Y} = \mathbf{1} \wedge \mathbf{Z} = \mathbf{0} \longrightarrow P$ (precondition strengthening)
  - $P \wedge \neg(Z \neq X) \longrightarrow \mathbf{Y} = \mathbf{X}!$ (postcondition weakening)

WHILE Rule - Loop invariant for factorial

$$\{P \wedge Z \neq X\}\ \mathtt{Z} := \mathtt{Z} + 1\ ;\ \mathtt{Y} := \mathtt{Y} \times \mathtt{Z}\ \{P\}$$

$$\mathbf{Y = 1 \wedge Z = 0} \longrightarrow P$$

$$P \wedge \neg(Z \neq X) \longrightarrow \mathbf{Y = X!}$$

- $\mathbf{Y = Z!}$
- VCs:
  - $\{\mathbf{Y = Z!} \wedge Z \neq X\}\ \mathtt{Z} := \mathtt{Z} + 1\ ;\ \mathtt{Y} := \mathtt{Y} \times \mathtt{Z}\ \{\mathbf{Y = Z!}\}$
    because: $\mathbf{Y = Z!} \wedge Z \neq X \longrightarrow Y \times (Z+1) = (Z+1)!$ and (10)
  - $\mathbf{Y = 1 \wedge Z = 0} \longrightarrow \mathbf{Y = Z!}$
    because: $0! = 1$
  - $\mathbf{Y = Z!} \wedge \neg(Z \neq X) \longrightarrow \mathbf{Y = X!}$
    because: $\neg(Z \neq X) \leftrightarrow Z = X$

$$\frac{\{Y \times (Z+1) = (Z+1)!\}\ \mathtt{Z} := \mathtt{Z} + 1\ \{Y \times Z = Z!\} \qquad \{Y \times Z = Z!\}\ \mathtt{Y} := \mathtt{Y} \times \mathtt{Z}\ \{\mathbf{Y = Z!}\}}{\{Y \times (Z+1) = (Z+1)!\}\ \mathtt{Z} := \mathtt{Z} + 1\ ;\ \mathtt{Y} := \mathtt{Y} \times \mathtt{Z}\ \{\mathbf{Y = Z!}\}} \quad (10)$$

WHILE Rule - Complete factorial example

$\{\mathbf{Y = 1 \wedge Z = 0}\}$

$\{\mathbf{Y = Z!}\}$

```
WHILE Z ≠ X DO
```

$\{\mathbf{Y = Z!} \wedge Z \neq X\}$

$\{Y \times (Z+1) = (Z+1)!\}$

```
   Z := Z + 1 ;
```

$\{Y \times Z = Z!\}$

```
   Y := Y × Z
```

$\{\mathbf{Y = Z!}\}$

```
OD
```

$\{\mathbf{Y = Z!} \wedge \neg Z \neq X\}$

$\{\mathbf{Y = X!}\}$

Another example - Multiplication!

$$\{\mathbf{Y} \geq \mathbf{0}\}$$

$$\{0 = (I - I) \times X\}$$

```
I := Y;
```

$$\{0 = (Y - I) \times X\}$$

```
Z := 0;
```

$$\{\mathbf{Z} = (\mathbf{Y} - \mathbf{I}) \times \mathbf{X}\}$$

```
WHILE I ≠ 0 DO
```

$$\{\mathbf{Z} = (\mathbf{Y} - \mathbf{I}) \times \mathbf{X} \wedge I \neq 0\}$$

$$\{Z + X = (Y - (I - 1)) \times X\}$$

```
   Z := Z + X ;
```

$$\{Z = (Y - (I - 1)) \times X\}$$

```
   I := I - 1
```

$$\{\mathbf{Z} = (\mathbf{Y} - \mathbf{I}) \times \mathbf{X}\}$$

```
OD
```

$$\{\mathbf{Z} = (\mathbf{Y} - \mathbf{I}) \times \mathbf{X} \wedge \neg I \neq 0\}$$

$$\{\mathbf{Z} = \mathbf{X} \times \mathbf{Y}\}$$

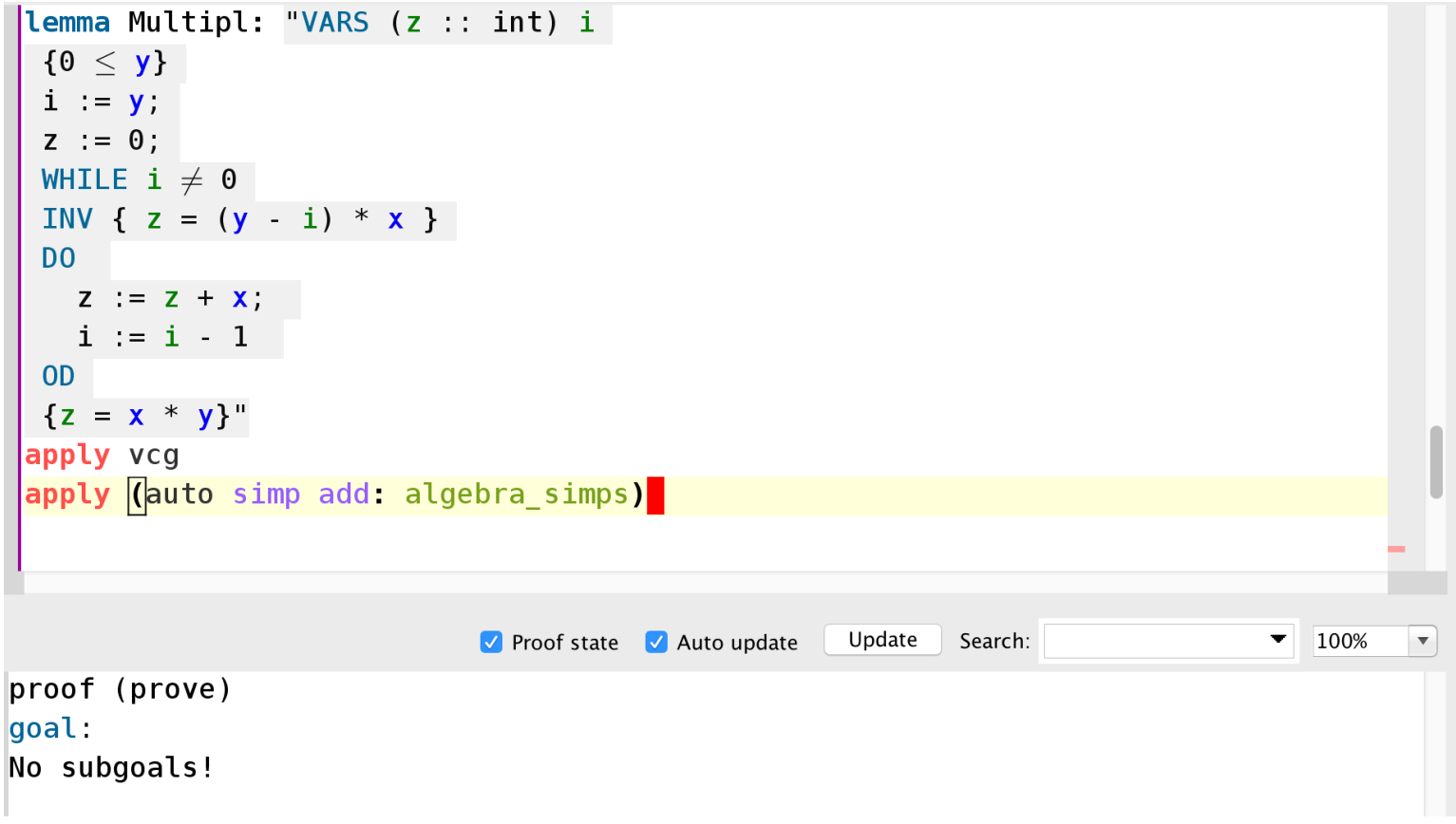
Isabelle

```
lemma Multipl: "VARS (z :: int) i
 {0 ≤ y}
 i := y;
 z := 0;
 WHILE i ≠ 0
 INV { z = (y - i) * x }
 DO
   z := z + x;
   i := i - 1
 OD
 {z = x * y}"
apply vcg
apply (auto simp add: algebra_simps)
```

Proof state Auto update Update Search: 100%

```
proof (prove)
goal (3 subgoals):
 1. ⋀z i. 0 ≤ y ⟹ 0 = (y - y) * x
 2. ⋀z i. z = (y - i) * x ∧ i ≠ 0 ⟹ z + x = (y - (i - 1)) * x
 3. ⋀z i. z = (y - i) * x ∧ ¬ i ≠ 0 ⟹ z = x * y
```

Isabelle


lemma Multipl: "VARS (z :: int) i
{0 ≤ y}
i := y;
z := 0;
WHILE i ≠ 0
INV { z = (y - i) * x }
DO
z := z + x;
i := i - 1
OD
{z = x * y}"
apply vcg
apply (auto simp add: algebra_simps)
Proof state
Auto update
Update
Search:
100%
proof (prove)
goal:
No subgoals!

Specification and correctness

$\{\mathbf{T}\}$
I := Y;
Z := 0;
WHILE I $\neq$ 0 DO
  Z := Z + X ;
  I := I - 1
OD
$\{\mathbf{Z} = \mathbf{X} \times \mathbf{Y}\}$

vs.

$\{\mathbf{Y} \geq \mathbf{0}\}$
I := Y;
Z := 0;
WHILE I $\neq$ 0 DO
  Z := Z + X ;
  I := I - 1
OD
$\{\mathbf{Z} = \mathbf{X} \times \mathbf{Y}\}$

- What is the difference? - *Termination!*

Hoare Logic Rules (it does!)

$$\frac{P \longrightarrow P' \quad \{P'\}\ C\ \{Q\}}{\{P\}\ C\ \{Q\}}\ PS \qquad \frac{\{P\}\ C\ \{Q'\} \quad Q' \longrightarrow Q}{\{P\}\ C\ \{Q\}}\ PW$$

$$\frac{}{\{Q[E/V]\}\ \mathtt{V} := \mathtt{E}\ \{Q\}}\ ASSIGN \qquad \frac{}{\{P\}\ \mathtt{SKIP}\ \{P\}}\ SKIP$$

$$\frac{\{P\}\ C_1\ \{Q\} \quad \{Q\}\ C_2\ \{R\}}{\{P\}\ C_1\ ;\ C_2\ \{R\}}\ SEQ$$

$$\frac{\{P \wedge S\}\ C_1\ \{Q\} \quad \{P \wedge \neg S\}\ C_2\ \{Q\}}{\{P\}\ \mathtt{IF}\ S\ \mathtt{THEN}\ C_1\ \mathtt{ELSE}\ C_2\ \mathtt{FI}\ \{Q\}}\ IF$$

$$\frac{\{P \wedge S\}\ C\ \{P\}}{\{P\}\ \mathtt{WHILE}\ S\ \mathtt{DO}\ C\ \mathtt{OD}\ \{P \wedge \neg S\}}\ WHILE$$

Other topics

$$\{P\}\ C\ \{Q\}$$

- Weakest preconditions, strongest postconditions

Other topics

$$\{P\}\ C\ \{Q\}$$

- Meta-theory: Is Hoare logic…
  - … *sound*? - Yes! Based on programming language semantics (but what about more complex languages?)
  - … *decidable*? - No! $\{\mathbf{T}\}\ C\ \{\mathbf{F}\}$ is the halting problem!
  - … *complete*? - *Relatively* / only for simple languages

Other topics

$$\{P\}\ C\ \{Q\}$$

- Automatic Verification Condition Generation (VCG)
- Automatic generation/inference of loop invariants!
- More complex languages - e.g. Pointers = Separation logic
- Functional programming (recursion = induction)

Another example

$\{\mathbf{T}\}$

$\{[\,] = rev(take\ 0\ A)\}$

```
j := 0; R := [ ];
```

$\{R = rev(take\ j\ A)\}$

```
WHILE j < length A DO
```

$\{R = rev(take\ j\ A) \wedge j < length\ A\}$

? ? ?

$\{A[j]\#R = rev(take\ (j+1)\ A)\}$

```
  R := A[j] # R ;
```

$\{R = rev(take\ (j+1)\ A)\}$

```
  j := j + 1
```

$\{R = rev(take\ j\ A)\}$

```
OD
```

$\{R = rev(take\ j\ A) \wedge \neg j < length\ A\}$

$\{\mathbf{R = rev\ A}\}$

Summary

- Precondition strengthening
- Postcondition weakening
- Automated generation of *Verification Conditions* (VCs)
- WHILE rule: *Loop invariants!*
  - Properties that hold during *while* loops
  - Loop invariant generation is generally *undecidable*

Recommended reading

Theory:

- Mike Gordon, *Background Reading on Hoare Logic*, `http://www.cl.cam.ac.uk/~mjcg/Teaching/2011/Hoare/Notes/Notes.pdf` (pp. 1-27, 37-48)
- Huth & Ryan, Sections 4.1-4.3 (pp. 256-292)
- Nipkow & Klein, Section 12.2.1 (pp. 191-199)

Practice:

- Isabelle's Hoare Logic library: `http://isabelle.in.tum.de/dist/library/HOL/HOL-Hoare`
- Tutorial exercise