BI as an Assertion Language for Mutable Data Structures

Samin Ishtiaq and Peter W. O'Hearn

Queen Mary & Westfield College, London

POPL 2001

(survey)

Introduction

Theme: Verification of programs

Approach: Hoare logic

Problem: Mutable data stuctures

Solution: Assertions use BI

Details:

1. Based on classical logic
2. Soundness and completeness
3. Local reasoning

Outline

1. Propositional BI
2. Pointer Model
3. Pre- and Postconditions (Syntax / Semantics / Properties)
4. Target Language (Syntax / Semantics)
5. Hoare Logic (+ Example)
6. New Axioms (+ Examples)
7. Soundness
8. Completeness
9. Locality of Specifications ans Reasoning
10. Related Works

Propositional BI - Proof-theoretic View

Connectives:

- Additive: $\bot$, $\rightarrow$($,\ \top,\ \vee,\ \wedge$)
- Multiplicative: $-\!\!*$, $*$, I

Bunches: $\Gamma ::= \phi \mid \{\}_m \mid \Gamma, \Gamma \mid \{\}_a \mid \Gamma; \Gamma$

Judgements: $\Gamma \vdash \phi$

Inference rules: $\ldots$

Strong normalization, Subject reduction property, Conservation over IL, Conservation over MILL, $\ldots$

Propositional BI - Resource Interpretation

Propositions with additive connectives:

$\phi \rightarrow \psi$: if I were to obtain enough money to make ϕ true, then I should also have enough to make ψ true

$\phi \wedge \psi$: the money I have got is both enough to make ϕ true and enough to make ψ true

$\phi \vee \psi$: the money I have got is enough either to make ϕ true or to make ψ true

Propositions with multiplicative connectives:

$\phi \mathrel{-\!\!*} \psi$: if you were to give me enough to make ϕ true then, combined with what I have already in my pocket, I should have enough to make ψ true

$\phi * \psi$: I can use part of my money to make ϕ true and have enough left over to make ψ true (anc vice versa)

Propositional BI - Model-theoretic View

A (preorder commutative) monoid $\mathcal{M} = (M, \cdot, e, \sqsubseteq)$

Clauses for the additives:

$$m \models \top \text{ iff always}$$

$$m \models \bot \text{ iff never}$$

$$m \models \phi \wedge \psi \text{ iff } m \models \phi \text{ and } m \models \psi$$

$$m \models \phi \vee \psi \text{ iff } m \models \phi \text{ or } m \models \psi$$

$$m \models \phi \to \psi \text{ iff for all } n \sqsubseteq m . n \models \phi \text{ implies } n \models \psi$$

Clauses for the multiplicatives:

$$m \models I \text{ iff } m \sqsubseteq e$$

$$m \models \phi * \psi \text{ iff for some } n, n' \in M . m \sqsubseteq n \cdot n' . n \models \phi \text{ and } n' \models \psi$$

$$m \models \phi \mathrel{-\!\!*} \psi \text{ iff } \forall n \in M . n \models \phi \text{ implies } n \cdot m \models \psi$$

Propositional BI - Comparisons

Basic propositions: *coin* (I have (at least) one coin in my pocket); *choc* (I have enough to buy a chocolate); *candy* (I have enough to buy a candy)

Axioms: $coin \vdash candy$; $coin * coin \vdash choc$

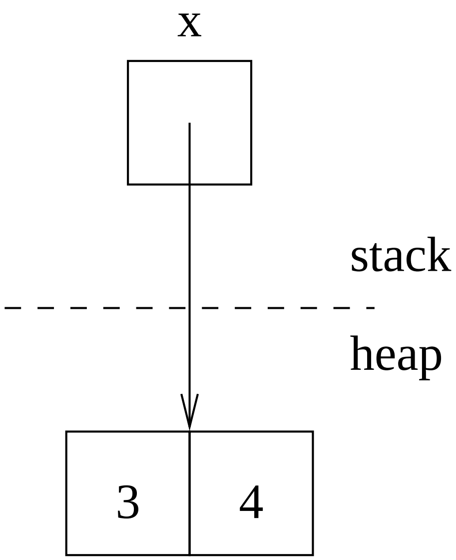
BI and Intuitionistic Logic

BI	IL
$coin \vdash coin \mathrel{-\!\!*} choc$	$coin \nvdash coin \to choc$

BI and Linear Logic

BI	LL
$coin \mathrel{-\!\!*} choc \nvdash coin \to choc$	$coin \multimap choc \vdash coin \to choc$

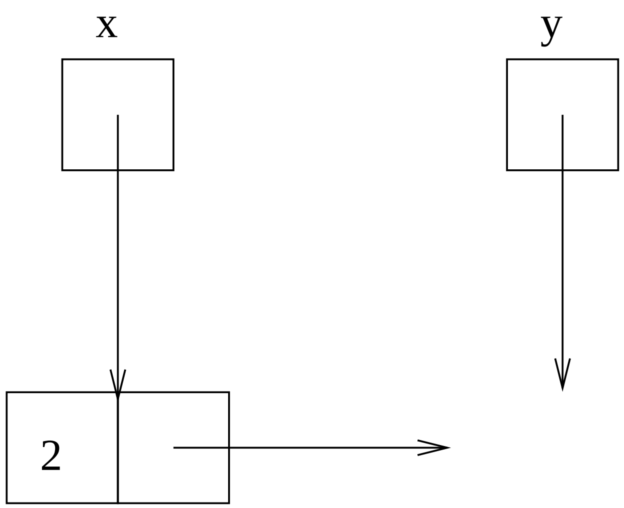
Pointer Model - States



$$\overbrace{[x \mapsto l]}^{s}, \overbrace{[l \mapsto \langle 2, l' \rangle]}^{h} \models \overbrace{(x \mapsto 3, 4)}^{P}$$

("P is true of the state s, h")

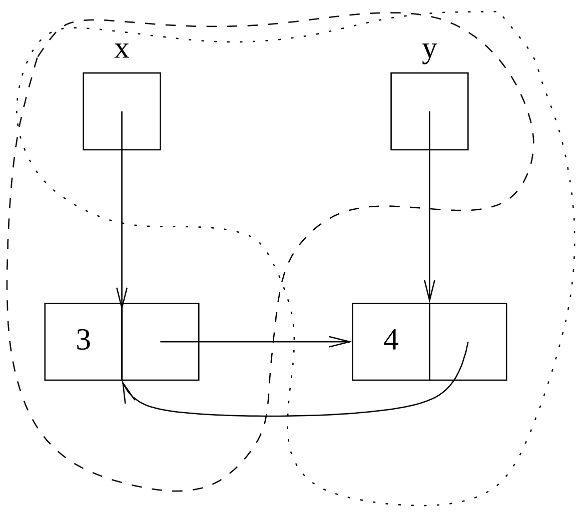
Pointer Model - Dangling Pointers



$$\overbrace{[x \mapsto l, y \mapsto l']}^{s}, \overbrace{[l \mapsto \langle 2, y\rangle]}^{h} \models (x \mapsto 2, y)$$

(the location l' is dangling)

Pointer Model - Spatial Conjunction



$$s, h \vDash (x \mapsto 3, y) * (y \mapsto 4, x)$$

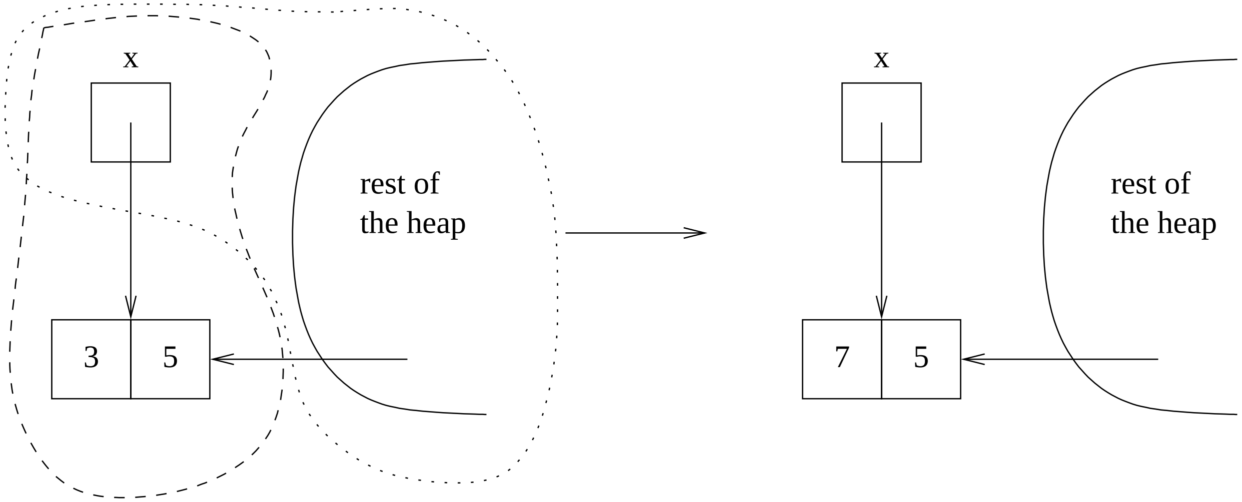
Pointer Model - Conjunctions

Difference between $*$ and $\wedge$:

1. $(x \mapsto 1, 2) * (x \mapsto 1, 2)$ is never true
2. $(x \mapsto 1, 2) \wedge (x \mapsto 1, 2) \equiv x \mapsto 1, 2$
3. $(x \mapsto 1, 2) * \neg(x \mapsto 1, 2)$ can be true
4. $(x \mapsto 1, 2) \wedge \neg(x \mapsto 1, 2)$ is never true

Pointer Model - Spatial Implication

$$s, h \models (x \mapsto 3, 5) * ((x \mapsto 7, 5) \mathrel{-\!\!*} P)$$



Pointer Model - Existential Quantification

A location in the heap is defined:

$$\exists ab.x \mapsto a, b$$

Pre- and Postconditions - Syntax

$$
\begin{array}{rcll}
P, Q, R & ::= & \alpha & \text{atomic formulae} \\
& \mid & \mathsf{false} & \text{falsity} \\
& \mid & P \to Q & \text{classical implication} \\
& \mid & \mathsf{emp} & \text{empty heap} \\
& \mid & P * Q & \text{spatial conjunction} \\
& \mid & P \mathrel{-\!\!*} Q & \text{spatial implication} \\
& \mid & \exists x.P & \text{existensial quantification}
\end{array}
$$

$$
\begin{array}{rcll}
\alpha & ::= & E = E' & \text{equality} \\
& \mid & E \mapsto E_1, E_2 & \text{points to} \\
& \mid & \vdots &
\end{array}
$$

Pre- and Postconditions - Semantics (1/2)

$s, h \models E = E'$	iff	$[\![E]\!]s = [\![E']\!]s$
$s, h \models E \mapsto E_1, E_2$	iff	$\{[\![E]\!]s\} = dom(h)$ and $h([\![E]\!]s) = \langle [\![E_1]\!]s, [\![E_2]\!]s \rangle$
$s, h \models \mathsf{emp}$	iff	$h = []$ is the empty heap
$s, h \models P * Q$	iff	$\exists h_0, h_1 . h_0 \# h_1, h_0 \cdot h_1 = h, s, h_0 \models P$ and $s, h_1 \models Q$
$s, h \models P \mathrel{-\!\!*} Q$	iff	$\forall h'.$if $h' \# h$ and $s, h' \models P$ then $, h \cdot h' \models Q$
$s, h \models \mathsf{false}$	iff	never
$s, h \models P \rightarrow Q$	iff	if $s, h \models P$ then $s, h \models Q$
$s, h \models \exists x.P$	iff	$\exists v \in Val.[s \mid x \mapsto v], h \models P$

Pre- and Postconditions - Semantics (2/2)

The points-to relation is 'exact': the domain of the heap must be a singleton.

'Inexact' points-to relation:

$$E \hookrightarrow E_1, E_2 = (\mathsf{true} * E \mapsto E_1, E_2)$$

Difference:

1. $P * (x \mapsto 1, 2) \rightarrow (x \mapsto 1, 2)$ is not always true
2. $P * (x \hookrightarrow 1, 2) \rightarrow (x \hookrightarrow 1, 2)$ is always true

Pre- and Postconditions - Properties (1/2)

Definition 1 (Consequence Relation) $P \models Q$ holds iff for all $s, h \models P$ then $s, h \models Q$, with $\mathrm{dom}(s) \supseteq \mathit{free}(P) \cup \mathit{free}(Q)$.

Proposition 1 The usual rules of classical logic are sound for $\models$, $*$ is commutative ans associative, with unit emp, and

$$\frac{P' \models P \quad Q' \models Q}{P' * Q' \models P * Q}$$

$$\frac{R * P \models Q}{R \models P \mathrel{-\!\!*} Q} \qquad \frac{R \models P \mathrel{-\!\!*} Q \quad R' \models P}{R * R' \models Q}$$

Pre- and Postconditions - Properties (2/2)

Contraction fails: $x \hookrightarrow 2, 3 \not\models (x \hookrightarrow 2, 3) * (x \hookrightarrow 2, 3)$ (because of #)

Weakening fails: $(x \mapsto 2, 3) * (y \mapsto 4, 5) \not\models (x \mapsto 2, 3)$ (because of the exact points-to relation)

Target Language -Syntax

$$
\begin{array}{rcl}
C & ::= & x := E \\
 & | & x := E.i \text{ where } i ::= 1|2 \\
 & | & E.i := E' \\
 & | & x := \mathsf{cons}(E_1, E_2) \\
 & | & \vdots
\end{array}
$$

$$
\begin{array}{rcll}
E & ::= & x & \text{variable} \\
 & | & 42 & \text{integer} \\
 & | & \mathsf{nil} & \text{nil} \\
 & | & a & \text{atom} \\
 & | & \vdots &
\end{array}
$$

Target Language - Operational Semantics

$$\frac{[\![E]\!]s = v}{x := E, s, h \rightsquigarrow [s|x \mapsto v], h}$$

$$\frac{[\![E]\!]s = l \in Loc \quad h(l) = r}{x := E.i, s, h \rightsquigarrow [s|x \mapsto \pi_i r], h}$$

$$\frac{[\![E]\!]s = l \in Loc \quad h(l) = r \quad [\![E']\!]s = v'}{E.i := E', s, h \rightsquigarrow s, [h|l \mapsto (r|i \mapsto v')]}$$

$$\frac{l \in Loc \quad l \notin dom(h) \quad [\![E_1]\!]s = v_1, [\![E_2]\!]s = v_2}{x := \mathsf{cons}(E_1, E_2), s, h \rightsquigarrow [s|x \mapsto l], [h|l \mapsto \langle v_1, v_2 \rangle]}$$

$$\frac{l \in Loc \quad l \in dom(h) \quad [\![E]\!]s = l}{\mathsf{dispose}(E), s, h \rightsquigarrow s, h - l}$$

Target Language - Run-time Error

Run-time error: dereference nil or an integer or a dangling pointer

- C, s, h is stuck when there is no configuration K such that $C, s, h \leadsto K$
- C, s, h is safe when $C, s, h \leadsto^* K$ implies that K is a terminal configuration or is not stuck.

Hoare Logic - Judgment

Definition 2 ((Partial Correctness) Interpretation)
$\{P\}C\{Q\}$ is true iff if $s, h \vDash P$ then C, s, h is safe and if $C, s, h \rightsquigarrow^* s', h'$ then $s', h' \vDash Q$ for all s, h where $dom(s) \supseteq free(P) \cup free(Q)$.

In practice, we try to prove $\{P\}C\{\mathsf{true}\}$ to conclude that C is safe to execute in any state satisfying P

Hoare Logic - Hoare 'Axioms'

Sequencing

$$\frac{\{P\}C\{Q\} \quad \{Q\}C'\{R\}}{\{P\}C;C'\{R\}}$$

Consequence

$$\frac{P \models P' \quad \{P'\}C\{R'\} \quad R' \models R}{\{P\}C\{R\}}$$

Simple Assignment

$$\{P[E/x]\}x := E\{P\}$$

While-loop

$$\frac{\{P \wedge Q\}c\{P\}}{\{P\}\text{while } Q \text{ do } c\{P \wedge \neg Q\}}$$

Hoare Logic - Example (1/3)

Does $w \equiv (\text{while } X > 0 \text{ do } Y := X \times Y; X := X - 1)$ compute factorials?

Formally, is $\{X = n \wedge n \geq 0 \wedge Y = 1\}w\{Y = n!\}$ provable?

We instantiate the rule for while-loops:

$$\frac{\{I \wedge X > 0\}c\{I\}}{\{I\}\text{while } X > 0 \text{ do } c\{I \wedge X \not> 0\}}$$

We need to:

1. Prove the premise (i.e., prove invariance)
2. Prove that the conclusion implies the desired property
3. Conclude by the consequence rule

We choose $I \equiv (Y \times X! = n! \wedge X \geq 0)$

Hoare Logic - Example (2/3)

We have chosen $I \equiv (Y \times X! = n! \wedge X \geq 0)$

Is $\{I \wedge X > 0\} Y := X \times Y; X := X - 1 \{I\}$ derivable?

By the rule of assignment: $\{I[(X-1)/X]\} X := X - 1 \{I\}$

$I[(X-1)/X] \equiv (Y \times (X-1)! = n! \wedge (X-1) \geq 0)$

By the rule of assignment:
$\{X \times Y \times (X-1)! = n! \wedge (X-1) \geq 0\} Y := X \times Y \{I[(X-1)/X]\}$

By the rule of sequencing:
$\{X \times Y \times (X-1)! = n! \wedge (X-1) \geq 0\} Y := X \times Y; X := X - 1 \{I\}$

$$
\begin{array}{rcl}
I \wedge X > 0 & \rightarrow & Y \times X! = n! \wedge X \geq 0 \wedge X > 0 \\
 & \rightarrow & Y \times X! = n! \wedge X \geq 1 \\
 & \rightarrow & Y \times X \times (X-1)! = n! \wedge (X-1) \geq 0
\end{array}
$$

We conclude by the rule of consequence.

Hoare Logic - Example (3/3)

We have chosen $I \equiv (Y \times X! = n! \wedge X \geq 0)$

By the rule for while loops: $\{I\}w\{I \wedge X \not> 0\}$

$$
\begin{array}{rcl}
I \wedge X \not> 0 & \rightarrow & Y \times X! = n! \wedge X \geq 0 \wedge X \not> 0 \\
& \rightarrow & Y \times X! = n! \wedge X = 0 \\
& \rightarrow & Y \times 0! = n! \\
& \xrightarrow{def} & Y = n!
\end{array}
$$

By the rule of consequence: $\{I\}w\{Y = n!\}$

$X = n \wedge n \geq 0 \wedge Y = 1 \rightarrow I$

By the rule of consequence:
$\{X = n \wedge n \geq 0 \wedge Y = 1\}w\{Y = n!\} \quad \Box$

Reynolds 'Axioms'

Object-component Lookup	$\{\exists x_1.P[x_1/x] \wedge \exists x_2.E \hookrightarrow x_1, x_2\}$ $x := E.1$ $\{P\}$
Object-component Assignment	$\{\exists x_1, \ldots, x_m.(E \mapsto E_1, E_2) * P\}$ $E.1 := E'$ $\{\exists x_1, \ldots, x_m.(E \mapsto E', E_2) * P\}$
Cons	$\{\exists x_1, \ldots, x_m.P\}$ $x := \mathsf{cons}(E_1, E_2)$ $\{\exists x', x_1, \ldots, x_m.P' * (x \mapsto E_1', E_2')\}$

Reynolds 'Axioms' - Note about Dangling

$$\{\mathsf{true}\}x := \mathsf{cons}(1,2)\{\neg(x = y)\}$$

is not true because of dangling.

Yet, it is not derivable by the cons axiom:

$$\{\mathsf{true}\}x := \mathsf{cons}(1,2)\{\underbrace{\mathsf{true} * x \mapsto 1,2}_{x \hookrightarrow 1,2}\}$$

Reynolds 'Axioms' - Example

Insertion of a cell in the middle of a linked list:

$$\{(x \mapsto a, z) * (y \mapsto c, w)\}$$

$$t := \mathsf{cons}(b, y)$$

$$\{(x \mapsto a, z) * (y \mapsto c, w) * (t \mapsto b, y)\}$$

$$\{(x \mapsto a, z) * (t \mapsto b, y) * (y \mapsto c, w)\}$$

$$x.2 := t$$

$$\{(x \mapsto a, t) * (t \mapsto b, y) * (y \mapsto c, w)\}$$

The Dispose 'Axiom'

Dispose

$$\{P * \exists ab.(E \mapsto a, b)\}$$
$$\mathsf{dispose}(E)$$
$$\{P\}$$

The Dispose 'Axiom' - Example 1

A program that does a double dispose is not safe to execute for any start state:

$$\{\mathsf{false}\}$$

$$\{\mathsf{true} * \exists ab.(x \mapsto a, b) * \exists cd.(x \mapsto c, d)\}$$

$$dispose(x)$$

$$\{\mathsf{true} * \exists ab.(x \mapsto a, b)\}$$

$$dispose(x)$$

$$\{\mathsf{true}\}$$

Dispose - Example 2 (1/2)

A program for disposing a list:

$$w \equiv \mathsf{while}\ p \neq \mathsf{nil}\ \mathsf{do}$$
$$q := p; p := p.2; \mathsf{dispose}(q)$$

A predicate to say that a list p points to a list: $\{\mathsf{nclist}\ p\}$

Partial Correctness Assertion: $\{\mathsf{nclist}\ p\}w\{\mathsf{emp}\}$

nclist and rep predicate definitions:

$\{\mathsf{nclist}\ E\} \equiv \exists n.\mathsf{rep}\ n\ E$

$\mathsf{rep}\ 0\ E \equiv E = \mathsf{nil} \wedge \mathsf{nil}$

$\mathsf{rep}\ n + 1\ E \equiv \exists xy.(E \mapsto x, y) * \mathsf{rep} n\ y$

Invariant of the while loop: $\{\mathsf{nclist}\ p\}$

Dispose - Example 2 (2/2)

Proof of invariance:

$\{\mathsf{nclist}\ p \wedge p \neq \mathsf{nil}\}$

$\downarrow$ *def*

$\{\exists x p_0.(p \mapsto x, p_0) * \mathsf{nclist}\ p_0\}$

$\downarrow$ (consequence rule)

$\{\exists p_0.(\mathsf{nclist}\ p_0 * \exists ab.(p \mapsto a, b)) \wedge \exists x.p \hookrightarrow x, p_0\}$

$q := p$

$\{\exists p_0.(\mathsf{nclist}\ p_0 * \exists ab.(q \mapsto a, b)) \wedge \exists x.p \hookrightarrow x, p_0\}$

$p := p.2$

$\{\mathsf{nclist}\ p * \exists ab.(q \mapsto a, b)\}$

$\mathsf{dispose}(q)$

$\{\mathsf{nclist}\ p\}$

By the while-loop rule: $\{\mathsf{nclist}\ p \wedge p = \mathsf{nil}\} \xrightarrow{\text{(consequence rule)}} \{\mathsf{emp}\} \quad \Box$

Hoare Logic - Properties

Soundness

Soundness means that all the triples that can be derived are valid partial correctness assertions.

Completeness

Completeness means that all the valid partial correctness assertions are triples that can be derived.

Completeness proof:

1. Turn axioms into 'backwards axioms'

2. Prove a lemma about weakest preconditions

3. Completeness proof

Backward 'Axioms'

Forward	**Backward**
$\{\exists x_1, \ldots, x_m.(E \mapsto E_1, E_2) * P\}$	$\{\exists xy.(E \mapsto x, y) * ((E \mapsto E', y) \mathrel{-\!\!*} P)\}$
$E.1 := E'$	$E.1 := E'$
$\{\exists x_1, \ldots, x_m.(E \mapsto E', E_2) * P\}$	$\{P\}$
$\{\exists x_1, \ldots, x_m.P\}$	$\{\forall x'.(x' \mapsto E_1, E_2) \mathrel{-\!\!*} P[x'/x]\}$
$x := \mathsf{cons}(E_1, E_2)$	$x := \mathsf{cons}(E_1, E_2)$
$\{\exists x', x_1, \ldots, x_m.P' * (x \mapsto E_1', E_2')\}$	$\{P\}$

Soundness

Soundness of backwards component assignment:

$$\{\exists xy.(E \mapsto x, y) * ((E \mapsto E', y) \mathrel{-\!\!*} P)\}$$
$$E.1 := E'$$
$$\{P\}$$

Suppose that s, h satisfies the precondition:

$$s, h \models (E \mapsto x, y) * ((E \mapsto E', y) \mathrel{-\!\!*} P)$$

It tells that there is a location $l \in dom(h)$ such that $h(l) = \langle x, y \rangle$. The execution of the assignment does not get stuck and after execution, the state becomes $s, [h|l \mapsto \langle [\![E']\!]s, y \rangle]$. We therefore need to show that $s, [h|l \mapsto \langle [\![E']\!]s, y \rangle] \models P$.

From the definition of $*$,:

$$s, [l] \models (E \mapsto x, y)$$
$$s, h - l \models (E \mapsto E', y) \mathrel{-\!\!*} P$$

The semantics of $\mathrel{-\!\!*}$ then implies that
$s, (h - l) \cdot [l \mapsto \langle [\![E']\!] s, y \rangle] \models P \quad \square$

Weakest Precondition

$$\{y \leq 3\} x := 2 * y + 1 \{x \leq 7\}$$

$$\{y = 1 \vee y = 3\} x := 2 * y + 1 \{x \leq 7\}$$

$$\{y = 1 \vee y = 3\} \rightarrow \{y \leq 3\}$$

DEFINITION 3 (WEAKEST PRECONDITION) $s, h \in wp(C, Q)$ iff C, s, h is safe and if $C, s, h \rightsquigarrow^* s', h'$ then $s', h' \vDash Q$

Lemma

THEOREM 1 The weakest precondition for each atomic statement is expressed by the corresponding backwards axiom, i.e. $\{w(C, Q)\}C\{Q\}$ is true.

'Completeness' of backwards component assignment:

$$\{\exists xy.(E \mapsto x, y) * ((E \mapsto E', y) \mathrel{-\!\!*} P)\}$$

$$E.1 := E'$$

$$\{P\}$$

Assume that $s, h \in wp(E.i := E', P)$. By definition of the weakest precondition, we know that there is a location $l \in dom(h)$ such that, say, $h(l) = \langle x, y \rangle$. Therefore, we already have $s, [l] \models (E \mapsto x, y)$. By definition of the weakest precondition, we also know that $s, [h|l \mapsto \langle [\![E']\!]s, y \rangle] \models P$. Therefore, for all the stacks h'' such that $s, h'' \models (E \mapsto E', y)$ (there is in fact only one stack that verifies that), $s, (h - l) \cdot h'' \models P$ (since $h - l \# h''$). Thus, $s, h - l \models ((E \mapsto E', y) -\!\!* P)$. The semantics of $*$ does the rest. $\square$

Completeness

THEOREM 2 (COMPLETENESS) Whenever a sequence C of assignment is true, then $\{P\}C\{Q\}$ is derivable from the basic axioms, Sequencing, and Consequence

Suppose that $\models \{P\}C\{Q\}$ is a valid partial correctness assertion. For all the worlds s, h, if $s, h \models P$ and if $C, s, h \rightsquigarrow s', h'$, then $s', h' \models Q$.

By definition of the weakest precondition, for all the worlds $s, h \in w(C, Q)$, if $C, s, h \rightsquigarrow s', h'$, then $s', h' \models Q$, i.e. $\{w(C, Q)\}C\{Q\}$ is true.

By the lemma, $\vdash \{w(C, Q)\}C\{Q\}$.

For all the worls s, h, if $s, h \models P$ then $s, h \models w(C, Q)$. $\models P \rightarrow w(C, Q)$ (or $P \models w(C, Q)$ by the deduction theorem).

By the consequence rule, $\{P\}C\{Q\}$ is true.

Locality of Specifications and Reasoning

Frame Axiom Introduction

$$\frac{\{P\}C\{Q\}}{\{P * R\}C\{Q * R\}} \quad \mathit{ModifiesOnly}(C) \cap \mathit{free}(R) = \emptyset$$

Example:

$$\{\mathsf{nclist}\ p\}\mathit{DisposeList}(p)\{\mathsf{emp}\}$$

$$\mathit{ModifiesOnly}(\mathit{DisposeList}(p)) = \{p\}$$

$$\dfrac{\dfrac{\dfrac{\{\mathsf{nclist}\ p\}DL(p)\{\mathsf{emp}\}}{\{(\mathsf{nclist}\ p) * (\mathsf{nclist}\ q)\}DL(p)\{\mathsf{emp} * \mathsf{nclist}\ q\}}}{\{(\mathsf{nclist}\ p) * (\mathsf{nclist}\ q)\}DL(p)\{\mathsf{nclist}\ q\}} \qquad \{\mathsf{nclist}\ q\}DL(q)\{\mathsf{emp}\}}{\{(\mathsf{nclist}\ p) * (\mathsf{nclist}\ q)\}DL(p); DL(q)\{\mathsf{emp}\}}$$

Related Works

Burstall, *Some techniques for proving correctness of programs which alter data* (1972)

Reynolds, *Intuitionistic reasoning about shared mutable data structure* (2000)

O'Hearn, Reynolds, Yang, *Local reasoning about programs that alter data structures* (2001)

Cardelli, Gordon, *Anytime, Anywhere, modal logics for mobile ambients*, (2000)

Conclusion

We have seen:

- how BI spatial connectives can be used in Hoare logic
- discuss the soundness and the completeness of an extended Hoare logic
- see how it enables local and compositional reasoning

We have not discussed the intuitionistic variant of the system.

Other points in resource semantics:

- location
- ownership