PRECISE, INTUITIONISTIC, AND SUPPORTED ASSERTIONS IN SEPARATION LOGIC

John C. Reynolds

Department of Computer Science
Carnegie Mellon University
and
Department of Informatics
Edinburgh University

MFPS XXI

Birmingham, England
May 21, 2005

©2005 John C. Reynolds

States

In Hoare logic:

$$\begin{aligned} \text{Values} &= \text{Integers} \\ \text{States} &= \text{Stores} \\ \text{Stores} &= \text{Variables} \to \text{Values}. \end{aligned}$$

In Separation logic:

$$\begin{aligned} \text{Values} &= \text{Integers} \\ \text{States} &= \text{Stores} \times \text{Heaps} \\ \text{Stores} &= \text{Variables} \to \text{Values} \\ \text{Heaps} &= \bigcup_{A \overset{\text{fin}}{\subseteq} \text{Addresses}} (A \to \text{Values}) \\ &\quad \text{where Addresses} \subseteq \text{Integers}. \end{aligned}$$

Assertions

Standard predicate calculus:

$$\wedge \qquad \vee \qquad \neg \qquad \Rightarrow \qquad \forall \qquad \exists$$

plus:

- **emp** — The heap is empty:

$$s, h \vDash \mathbf{emp} \text{ iff } \operatorname{dom} h = \{\}.$$

- $e \mapsto e'$ — The heap contains one cell, at address e with contents e':

$$s, h \vDash e \mapsto e' \text{ iff } \operatorname{dom} h = \{[\![e]\!]_{\text{exp}} s\} \text{ and } h([\![e]\!]_{\text{exp}} s) = [\![e']\!]_{\text{exp}} s.$$

- $p_1 \ * \ p_2$ — The heap can be split into two disjoint parts such that p_1 holds for one part and p_2 holds for the other:

$$s, h \vDash p_0 \ * \ p_1 \text{ iff } \exists h_0, h_1.\ h_0 \perp h_1 \text{ and } h_0 \cdot h_1 = h \text{ and } s, h_0 \vDash p_0 \text{ and } s, h_1 \vDash p_1.$$

- $p_1 -\!\!* \ p_2$ — If the current heap is extended with a disjoint part in which p_1 holds, then p_2 holds for the extended heap:

$$s, h \vDash p_0 -\!\!* \ p_1 \text{ iff } \forall h'.\ (h' \perp h \text{ and } s, h' \vDash p_0) \text{ implies } s, h \cdot h' \vDash p_1.$$

Some Abbreviations

$$e \mapsto - \stackrel{\text{def}}{=} \exists x'.\ e \mapsto x' \quad \text{where } x' \text{ not free in } e$$

$$e \hookrightarrow e' \stackrel{\text{def}}{=} e \mapsto e' \ * \ \mathbf{true}$$

$$e \mapsto e_0, \ldots, e_{n-1} \stackrel{\text{def}}{=} e \mapsto e_0 \ * \ \cdots \ * \ e + n - 1 \mapsto e_{n-1}$$

$$e \hookrightarrow e_0, \ldots, e_{n-1} \stackrel{\text{def}}{=} e \hookrightarrow e_0 \ * \ \cdots \ * \ e + n - 1 \hookrightarrow e_{n-1}$$

$$\text{iff } e \mapsto e_0, \ldots, e_{n-1} \ * \ \mathbf{true}.$$

Precise Assertions

An assertion q is *precise* iff

For all s and h, there is at most one $h' \subseteq h$ such that

$$s, h' \vDash q$$

is true.

Examples of Precise Assertions

- $\mathbf{emp}$, $e \mapsto e'$, $e \mapsto -$.
- $p \ast q$, when p and q are precise.
- p when $p \Rightarrow q$ is valid and q is precise.
- $\mathsf{list}\ \alpha\ \mathsf{i}$ and $\exists \alpha.\ \mathsf{list}\ \alpha\ \mathsf{i}$, where α ranges over sequences of integers and

$$\mathsf{list}\ \epsilon\ \mathsf{i} \stackrel{\mathrm{def}}{=} \mathbf{emp} \wedge \mathsf{i} = \mathbf{nil}$$

$$\mathsf{list}\ (\mathsf{a}{\cdot}\alpha)\ \mathsf{i} \stackrel{\mathrm{def}}{=} \exists \mathsf{j}.\ \mathsf{i} \mapsto \mathsf{a}, \mathsf{j} \ast \mathsf{list}\ \alpha\ \mathsf{j}.$$

- $\mathsf{tree}\ \tau\ (i)$, and $\exists \tau.\ \mathsf{tree}\ \tau\ (i)$, where τ ranges over S-expressions and

$$\mathsf{tree}\ a\ (i) \stackrel{\mathrm{def}}{=} \mathbf{emp} \wedge i = a \qquad (a \text{ an atom})$$

$$\mathsf{tree}\ (\tau_1 \cdot \tau_2)\ (i) \stackrel{\mathrm{def}}{=}$$
$$\exists i_1, i_2.\ \ i \mapsto i_1, i_2 \ast \mathsf{tree}\ \tau_1\ (i_1) \ast \mathsf{tree}\ \tau_2\ (i_2).$$

Examples of Imprecise Assertions

- **true**
- $\mathbf{emp} \vee \mathsf{x} \mapsto 10$
- $\mathsf{x} \mapsto 10 \vee \mathsf{y} \mapsto 10$
- $\exists \mathsf{x}.\ \mathsf{x} \mapsto 10$

Notice that

$$[\mathbf{emp}]$$
$$\mathsf{x} := \mathbf{cons}(10)\ ;\ \mathsf{x} := \mathbf{nil}$$
$$[\exists \mathsf{x}.\ \mathsf{x} \mapsto 10]$$

We will see that the impreciseness of the postcondition here is linked to the existence of a memory leak.

More Examples

- $\mathsf{lseg}\ \alpha\ (\mathsf{i}, \mathsf{j})$ is precise, but $\exists \alpha.\ \mathsf{lseg}\ \alpha\ (\mathsf{i}, \mathsf{j})$ is imprecise, where

$$\mathsf{lseg}\ \epsilon\ (\mathsf{i}, \mathsf{j}) \stackrel{\text{def}}{=} \mathbf{emp} \wedge \mathsf{i} = \mathsf{j}$$
$$\mathsf{lseg}\ \mathsf{a}{\cdot}\alpha\ (\mathsf{i}, \mathsf{k}) \stackrel{\text{def}}{=} \exists \mathsf{j}.\ \mathsf{i} \mapsto \mathsf{a}, \mathsf{j}\ *\ \mathsf{lseg}\ \alpha\ (\mathsf{j}, \mathsf{k}).$$

For instance, if

$$s(\mathsf{i}) = s(\mathsf{j}) = 1 \qquad \mathrm{dom}\, h = \{1, 2\} \qquad h(1) = 10 \qquad h(2) = 1,$$

then both h and the empty heap satisfy $\mathsf{lseg}\ \alpha\ (\mathsf{i}, \mathsf{j})$.

On the other hand,

- Both $\mathsf{ntlseg}\ \alpha\ (\mathsf{i}, \mathsf{j})$ and $\exists \alpha.\ \mathsf{ntlseg}\ \alpha\ (\mathsf{i}, \mathsf{j})$ are precise, where

$$\mathsf{ntlseg}\ \epsilon\ (\mathsf{i}, \mathsf{j}) \stackrel{\text{def}}{=} \mathbf{emp} \wedge \mathsf{i} = \mathsf{j}$$
$$\mathsf{ntlseg}\ \mathsf{a}{\cdot}\alpha\ (\mathsf{i}, \mathsf{k}) \stackrel{\text{def}}{=} \mathsf{i} \neq \mathsf{k} \wedge \mathsf{i} \neq \mathsf{k} + 1 \wedge$$
$$(\exists \mathsf{j}.\ \mathsf{i} \mapsto \mathsf{a}, \mathsf{j}\ *\ \mathsf{ntlseg}\ \alpha\ (\mathsf{j}, \mathsf{k})).$$

Preciseness and Distributivity

The semi-distributive laws

$$(p_1 \land p_2) \ast q \Rightarrow (p_1 \ast q) \land (p_2 \ast q)$$
$$(\forall x.\, p) \ast q \Rightarrow \forall x.\, (p \ast q) \quad \text{when } x \text{ not free in } q$$

are valid for all assertions. But their converses

$$(p_1 \ast q) \land (p_2 \ast q) \Rightarrow (p_1 \land p_2) \ast q$$
$$\forall x.\, (p \ast q) \Rightarrow (\forall x.\, p) \ast q \quad \text{when } x \text{ not free in } q$$

are not. For example, when

$$s(\mathsf{x}) = 1 \quad s(\mathsf{y}) = 2 \quad \mathrm{dom}\, h = \{1, 2\} \quad h(1) = 10 \quad h(2) = 20,$$

the assertion

$$(\mathsf{x} \mapsto 10 \ast (\mathsf{x} \mapsto 10 \lor \mathsf{y} \mapsto 20)) \land (\mathsf{y} \mapsto 20 \ast (\mathsf{x} \mapsto 10 \lor \mathsf{y} \mapsto 20))$$

is true, but

$$((\mathsf{x} \mapsto 10 \land \mathsf{y} \mapsto 20) \ast (\mathsf{x} \mapsto 10 \lor \mathsf{y} \mapsto 20))$$

is false.

However, the converses are valid when q is precise.

Preciseness and Distributivity (continued)

When q is precise,

$$(p_1 \ * \ q) \wedge (p_2 \ * \ q) \Rightarrow (p_1 \wedge p_2) \ * \ q.$$

Proof: Suppose $s, h \vDash (p_1 \ * \ q) \wedge (p_2 \ * \ q)$. Then there are:

An $h_1 \subseteq h$ such that $s, h - h_1 \vDash p_1$ and $s, h_1 \vDash q$,

An $h_2 \subseteq h$ such that $s, h - h_2 \vDash p_2$ and $s, h_2 \vDash q$.

Then, since q is precise,

$$h_1 = h_2$$
$$h - h_1 = h - h_2$$
$$s, h - h_1 \vDash p_1 \wedge p_2$$
$$s, h \vDash (p_1 \wedge p_2) \ * \ q.$$

The Programming Language: An Informal View

The simple imperative language:

$:=$ **skip** ; **if** $-$ **then** $-$ **else** $-$ **while** $-$ **do** $-$

plus:

		Store : $\mathsf{x}\colon 3,\ \mathsf{y}\colon 4$ Heap : empty
Allocation	$\mathsf{x} := \mathbf{cons}(1, 2)$;	
		Store : $\mathsf{x}\colon 37,\ \mathsf{y}\colon 4$ Heap : $37\colon 1,\ 38\colon 2$
Lookup	$\mathsf{y} := [\mathsf{x}]$;	
		Store : $\mathsf{x}\colon 37,\ \mathsf{y}\colon 1$ Heap : $37\colon 1,\ 38\colon 2$
Mutation	$[\mathsf{x} + 1] := 3$;	
		Store : $\mathsf{x}\colon 37,\ \mathsf{y}\colon 1$ Heap : $37\colon 1,\ 38\colon 3$
Deallocation	$\mathbf{dispose}(\mathsf{x} + 1)$	
		Store : $\mathsf{x}\colon 37,\ \mathsf{y}\colon 1$ Heap : $37\colon 1$

Note that:

- Expressions depend only upon the store.
  - no side effects or nontermination.
  - **cons** and $[-]$ are parts of commands.
- Allocation is nondeterminate.

Memory Faults

		Store : $\mathsf{x}\colon 3,\ \mathsf{y}\colon 4$
		Heap : empty
Allocation	$\mathsf{x} := \mathbf{cons}(1, 2)\ ;$	
		Store : $\mathsf{x}\colon 37,\ \mathsf{y}\colon 4$
		Heap : $37\colon 1,\ 38\colon 2$
Lookup	$\mathsf{y} := [\mathsf{x}]\ ;$	
		Store : $\mathsf{x}\colon 37,\ \mathsf{y}\colon 1$
		Heap : $37\colon 1,\ 38\colon 2$
Mutation	$[\mathsf{x} + 2] := 3\ ;$	
		abort

Faults can also be caused by out-of-range lookup or deallocation.

Large-Step Semantics of Commands

$$[\![c \in \langle\text{comm}\rangle]\!] \subseteq \text{States} \times (\text{States} \cup \{\textbf{abort}, \bot\})$$

is the relation such that

- $(s, h)\, [\![c]\!]\, (s', h')$ iff, starting in state (s, h), there is a finite execution of c that terminates in state (s', h').
- $(s, h)\, [\![c]\!]\, \textbf{abort}$ iff, starting in state (s, h), there is a finite execution of c that terminates with **abort**.
- $(s, h)\, [\![c]\!]\, \bot$ iff, starting in state (s, h), there is a nonterminating execution of c.

Some Definitions

When $(s, h)\, [\![c]\!]\, \textbf{abort}$ is false, we say that c is *safe at* (s, h).

When $(s, h)\, [\![c]\!]\, \textbf{abort}$ and $(s, h)\, [\![c]\!]\, \bot$ are both false, we say that *c must terminate normally at* (s, h).

Totality

Our semantics is *total:* For every $(s, h) \in \mathrm{States}$, there is at least one $\tau' \in \mathrm{States} \cup \{\textbf{abort}, \bot\}$ such that $(s, h)\, [\![c]\!]\, \tau'$.

Thus, if c must terminate normally at (s, h), then there is at least one state (s', h') such that $(s, h)\, [\![c]\!]\, (s', h')$.

Safety Monotonicity

If $h_0 \subseteq h$ and c is safe at (s, h_0), then c is safe at (s, h).

If $h_0 \subseteq h$ and c must terminate normally at (s, h_0), then c must terminate normally at (s, h).

The Frame Property

If

$$h_0 \subseteq h,\ c \text{ is safe at } (s, h_0), \text{ and } (s, h)\, [\![c]\!]\, (s', h'),$$

then

$$h - h_0 \subseteq h' \text{ and } (s, h_0)\, [\![c]\!]\, (s', h' - (h - h_0)).$$

Specifications

Partial correctness:

$\{p\}\ c\ \{q\}$ *holds* iff $\forall(s,h) \in \mathrm{States}.\ s,h \vDash p$ implies

$\neg(s,h)\ [\![c]\!]_{\mathrm{comm}}$ **abort**

$\{$i.e., c is safe at $(s,h)\}$

and $\Big(\forall(s',h') \in \mathrm{States}.\ (s,h)\ [\![c]\!]_{\mathrm{comm}}\ (s',h')$ implies $s',h' \vDash q\Big)$.

Total correctness:

$[\ p\ ]\ c\ [\ q\ ]$ *holds* iff $\forall(s,h) \in \mathrm{States}.\ s,h \vDash p$ implies

$\neg(s,h)\ [\![c]\!]_{\mathrm{comm}}$ **abort** and $\neg(s,h)\ [\![c]\!]_{\mathrm{comm}}\ \bot$

$\{$i.e., c must terminate normally at $(s,h)\}$

and $\Big(\forall(s',h') \in \mathrm{States}.\ (s,h)\ [\![c]\!]_{\mathrm{comm}}\ (s',h')$ implies $s',h' \vDash q\Big)$.

The Frame Rule

$$\frac{\{p\}\ c\ \{q\}}{\{p\ *\ r\}\ c\ \{q\ *\ r\},}$$

when no variable occurs in both $\mathrm{FV}(r)$ and $\mathrm{MD}(c)$.

Preciseness and Hoare Triples

Suppose $[\,p\,]\; c\; [\,q\,]$ holds and q is precise. Then p is precise.

Proof: Suppose $[\,p\,]\; c\; [\,q\,]$, q is precise, and there are heaps $h_1, h_2 \subseteq h$ such that $s, h_1 \vDash p$ and $s, h_2 \vDash p$. We must show $h_1 = h_2$.

Since $[\,p\,]\; c\; [\,q\,]$ and $s, h_1 \vDash p$,

$$c \text{ must terminate normally at } s, h_1.$$

By safety monotonicity and $h_1 \subseteq h$:

$$c \text{ must terminate normally at } s, h.$$

By totality, there is a state s', h' such that

$$(s, h)[\![c]\!](s', h').$$

By the frame property, $h - h_1 \subseteq h'$ and

$$(s, h_1)[\![c]\!](s', h' - (h - h_1)).$$

Then, since $[\,p\,]\; c\; [\,q\,]$ and $s, h_1 \vDash p$,

$$s', h' - (h - h_1) \vDash q.$$

By a similar argument,

$$s', h' - (h - h_2) \vDash q.$$

Then, since q is precise and $h' - (h - h_1), h' - (h - h_2) \subseteq h'$,

$$\begin{aligned} h' - (h - h_1) &= h' - (h - h_2), \\ h - h_1 &= h - h_2, \\ h_1 &= h_2 \end{aligned}$$

Memory Leaks

We say that c_1 has a memory leak at p if there is no c_2 such that

$$[\, p \,]\; c_1 \,; c_2 \;[\, \mathbf{emp} \,]$$

holds.

Then c_1 has a memory leak at p if there is no precise q such that $[\, p \,]\; c_1 \;[\, q \,]$.

Proof: Suppose c_1 does not have a memory leak at p. Then there is a c_2 such that

$$[\, p \,]\; c_1 \,; c_2 \;[\, \mathbf{emp} \,].$$

Let q be the weakest precondition of c_2 for **emp**. Then

$$[\, p \,]\; c_1 \;[\, q \,] \qquad \text{and} \qquad [\, q \,]\; c_2 \;[\, \mathbf{emp} \,].$$

But **emp** is precise, so that q must be precise.

The Hypothetical Frame Rule

$$\frac{\Gamma, \{p_1\}\ k_1\ \{q_1\}[X_1], \ldots, \{p_n\}\ k_n\ \{q_n\}[X_n] \vdash \{p\}\ c\ \{q\}}{\Gamma, \{p_1\ *\ r\}\ k_1\ \{q_1\ *\ r\}[X_1], \ldots, \{p_n\ *\ r\}\ k_n\ \{q_n\ *\ r\}[X_n] \vdash \{p\ *\ r\}\ c\ \{q\ *\ r\},}$$

where ... (various restrictions on variables).

Theorem 5:

1. The hypothetical frame rule is sound for fixed preconditions $p_1, \ldots, p_n$ if and only if $p_1, \ldots, p_n$ are all precise.

2. The hypothetical frame rule is sound for a fixed invariant r if and only if r is precise.

(From O'Hearn, Yang, and Reynolds, "Separation and Information Hiding".)

Shared-Variable Concurrency with Critical Regions

- Region

$$\frac{\Gamma \vdash \{(p \ast R) \land b\}\ c\ \{q \ast R\}}{\Gamma, r(X) : R \vdash \{p\}\ \textbf{with}\ r\ \textbf{when}\ b\ \textbf{do}\ c\ \{q\}}$$

  if R precise.

- Resource

$$\frac{\Gamma, r(X) : R \vdash \{p\}\ c\ \{q\}}{\Gamma \vdash \{p \ast R\}\ \textbf{resource}\ r\ \textbf{in}\ c\ \{q \ast R\}}$$

  if R precise.

(Adapted from Brookes, “A Semantics for Concurrent Separation Logic”.)

Intuitionistic Assertions

- An assertion i is *intuitionistic* iff, for all stores s and heaps h and h':

$$h \subseteq h' \text{ and } s, h \vDash i \text{ implies } s, h' \vDash i.$$

In the following, we assume i and i' are intuitionistic assertions, p is any assertion, and e and e' are expressions:

- Syntactically, the following assertions are intuitionistic:

$$\begin{array}{cc} \text{Any pure assertion} & p \ast i \\ p \mathrel{-\!\!\ast} i & i \mathrel{-\!\!\ast} p \\ i \wedge i' & i \vee i' \\ \forall v.\ i & \exists v.\ i, \end{array}$$

  and as special cases:

$$p \ast \mathbf{true} \qquad \mathbf{true} \mathrel{-\!\!\ast} p \qquad e \hookrightarrow e'.$$

- The following laws hold:

$$i \ast p \Rightarrow i \qquad i \Rightarrow p \mathrel{-\!\!\ast} i \qquad i \ast i' \Rightarrow i \wedge i'.$$

- $p \ast \mathbf{true}$ is the strongest intuitionistic assertion weaker than p.

- $\mathbf{true} \mathrel{-\!\!\ast} p$ is the weakest intuitionistic assertion that is stronger than p.

- No satisfiable assertion is both precise and intuitionistic.

Supported Assertions

- An assertion q is *supported* iff, for all s, h_1, and h_2,
  - if $h_1 \cup h_2$ is a function, and $s, h_1 \models q$ and $s, h_2 \models q$ are true,
  - then there is an h' such that $h' \subseteq h_1$, $h' \subseteq h_2$, and $s, h' \models q$ is true.

Equivalently,

- An assertion q is *supported* iff, for all s and h, if the set

$$H = \{\, h' \mid h' \subseteq h \text{ and } s, h' \models q \,\}$$

is nonempty, then it has a least element.

Examples

- Imprecise, Intuitionistic, and Supported
  - **true**
  - $\mathsf{x} \hookrightarrow 10$
  - $\mathsf{x} \hookrightarrow 10 \wedge \mathsf{y} \hookrightarrow 10$
  - $\mathsf{dag}\,\tau\,(i)$ and $\exists\tau.\ \mathsf{dag}\,\tau\,(i)$, where τ ranges over S-expressions and

$$\mathsf{dag}\,a\,(i) \stackrel{\text{def}}{=} i = a \qquad (a \text{ an atom})$$

$$\mathsf{dag}\,(\tau_1 \cdot \tau_2)\,(i) \stackrel{\text{def}}{=} \exists i_1, i_2.\ \ i \mapsto i_1, i_2 \ * \ (\mathsf{dag}\,\tau_1\,(i_1) \wedge \mathsf{dag}\,\tau_2\,(i_2)).$$

- Imprecise, Intuitionistic, Unsupported
  - $\mathsf{x} \hookrightarrow 10 \vee \mathsf{y} \hookrightarrow 10$
  - $\exists \mathsf{x}.\ \mathsf{x} \hookrightarrow 10$
  - $\neg$ **emp**
- Imprecise, Nonintuitionistic, Supported
  - **emp** $\vee\ \mathsf{x} \mapsto 10$
- Imprecise, Nonintuitionistic, Unsupported
  - $\mathsf{x} \mapsto 10 \vee \mathsf{y} \mapsto 10$
  - $\exists \mathsf{x}.\ \mathsf{x} \mapsto 10$

Supported Assertions and Distributivity

When q is supported and the remaining assertions are intuitionistic, the semidistributive laws given earlier become full distributive laws.

When p_1 and p_2 are intuitionistic and q is supported:

$$(p_1 \ * \ q) \wedge (p_2 \ * \ q) \Rightarrow (p_1 \wedge p_2) \ * \ q,$$

When p is intuitionistic, q is supported, and x is not free in q:

$$\forall x.\ (p \ * \ q) \Rightarrow (\forall x.\ p) \ * \ q.$$

Proof (of the first law): Suppose $s, h \vDash (p_1 \ * \ q) \wedge (p_2 \ * \ q)$. Then there are:

An $h_1 \subseteq h$ such that $s, h - h_1 \vDash p_1$ and $s, h_1 \vDash q$,

An $h_2 \subseteq h$ such that $s, h - h_2 \vDash p_2$ and $s, h_2 \vDash q$.

Then, since q is supported and $h_1 \cup h_2$ is a function, there is an $h' \subseteq h_1, h_2$ such that $s, h' \vDash q$. Moreover, since $h - h_1, h - h_2 \subseteq h - h'$, and p_1 and p_2 are intuitionistic,

$$s, h - h' \vDash p_1 \wedge p_2$$

$$s, h \vDash (p_1 \wedge p_2) \ * \ q.$$

The Hypothetical Frame Rule

$$\frac{\Gamma, \{p_1\}\ k_1\ \{q_1\}[X_1], \ldots, \{p_n\}\ k_n\ \{q_n\}[X_n] \vdash \{p\}\ c\ \{q\}}{\Gamma, \{p_1 \ *\ r\}\ k_1\ \{q_1 \ *\ r\}[X_1], \ldots, \{p_n \ *\ r\}\ k_n\ \{q_n \ *\ r\}[X_n] \vdash \{p \ *\ r\}\ c\ \{q \ *\ r\}},$$

where ... (various restrictions on variables).

Theorem 11: The hypothetical frame rule is sound in the following cases:

1. The preconditions $p_1, \ldots, p_n$ are supported, and the postconditions $q_1, \ldots, q_n$ are intuitionistic.

2. The resource invariant r is supported, and the postconditions $q_1, \ldots, q_n$ are intuitionistic.

(From O'Hearn, Yang, and Reynolds, "Separation and Information Hiding".)

Shared-Variable Concurrency with Critical Regions

- Region

$$\frac{\Gamma \vdash \{(p \ast R) \wedge b\}\ c\ \{q \ast R\}}{\Gamma, r(X) : R \vdash \{p\}\ \textbf{with}\ r\ \textbf{when}\ b\ \textbf{do}\ c\ \{q\}}$$

if R supported, p and q intuitionistic.

- Resource

$$\frac{\Gamma, r(X) : R \vdash \{p\}\ c\ \{q\}}{\Gamma \vdash \{p \ast R\}\ \textbf{resource}\ r\ \textbf{in}\ c\ \{q \ast R\}}$$

if R supported, p and q intuitionistic.

(From Brookes, “A Semantics for Concurrent Separation Logic”.)

The Operation $- * \textbf{true}$

- If p is any assertion, then $p * \textbf{true}$ is intuitionistic.
- If q is intuitionistic, then $q \Leftrightarrow q * \textbf{true}$.

Thus $- * \textbf{true}$ maps arbitrary assertions into intuitionistic assertions, and acts as an identity on the latter.

- If p is precise, then p is supported.
- If q is supported, then $q * \textbf{true}$ is supported.
- Therefore, if p is precise, the $p * \textbf{true}$ is supported.

Thus $- * \textbf{true}$ maps precise assertions into supported intuitionistic assertions, and acts as an identity on the latter.

The Precising Operation (Yang)

$$\mathrm{Pr}\, p \stackrel{\mathrm{def}}{=} p \wedge \neg(p \; * \; \neg\, \mathbf{emp}).$$

For example:

$$\mathrm{Pr}\, \mathbf{true} \;\text{ iff }\; \mathbf{emp}$$

$$\mathrm{Pr}\, \mathsf{x} \hookrightarrow 10 \;\text{ iff }\; \mathsf{x} \mapsto 10$$

$$\mathrm{Pr}\, (\mathsf{x} \hookrightarrow 10 \wedge \mathsf{y} \hookrightarrow 10) \;\text{ iff}$$
$$\mathbf{if}\ \mathsf{x} = \mathsf{y}\ \mathbf{then}\ \mathsf{x} \mapsto 10\ \mathbf{else}\ (\mathsf{x} \mapsto 10 \; * \; \mathsf{y} \mapsto 10)$$

$$\mathrm{Pr}\, (\mathbf{emp} \vee \mathsf{x} \mapsto 10 \text{ iff})\ \mathbf{emp}$$

and Pr $\mathsf{dag}\ \tau\ (i)$ and Pr $\exists\tau.\ \mathsf{dag}\ \tau\ (i)$, if satisfiable, both describe the heap containing the dag rooted at i and nothing else.

- If p is supported, then Pr p is precise.
- If p is precise, then Pr $p \Leftrightarrow p$.

Thus Pr maps supported operations into precise operations, and acts as an identity on the latter.

An Isomorphism

- $\Pr\,(p\ *\ \mathbf{true}) \Rightarrow p$.
- $p \Rightarrow \Pr\,(p\ *\ \mathbf{true})$ when p is precise.
- Therefore $\Pr\,(p\ *\ \mathbf{true}) \Leftrightarrow p$ when p is precise.
- $(\Pr\ q)\ *\ \mathbf{true} \Rightarrow q$ when q is intuitionistic.
- $q \Rightarrow (\Pr\ q)\ *\ \mathbf{true}$ when q is supported.
- Therefore $(\Pr\ q)\ *\ \mathbf{true} \Rightarrow q$ when q is supported and intuitionistic.

Thus $-\ *\ \mathbf{true}$ and $\Pr$ are isomorphisms between the set of precise assertions and the set of supported intuitionistic assertions.

The Good News

The inference rule

$$\frac{\{p\}\ c\ \{p'\}}{\{p\ *\ \mathbf{true}\}\ c\ \{p'\ *\ \mathbf{true}\}}$$

is sound (since it follows from the frame rule). Thus Hoare triples respect the operation $-\ *\ \mathbf{true}$.

The Bad News

The inference rule

$$\frac{\{p\}\ c\ \{p'\}}{\{\mathrm{Pr}\ p\}\ c\ \{\mathrm{Pr}\ p'\}}$$

is not sound. For example,

$$\{\mathbf{true}\}\ \mathsf{x} := \mathbf{cons}(10)\ \{\mathbf{true}\}$$

holds, but

$$\{\mathbf{emp}\}\ \mathsf{x} := \mathbf{cons}(10)\ \{\mathbf{emp}\}$$

does not hold. Thus Hoare triples do not respect Pr.