AN INTRODUCTION TO SEPARATION LOGIC

2. Assertions

John C. Reynolds
Carnegie Mellon University

January 7, 2011

Some Notation for Functions

We write

$$[\, x_1 \colon y_1 \mid \ldots \mid x_n \colon y_n \,]$$

for the function with domain $\{x_1, \ldots, x_n\}$ that maps each x_i into y_i, and

$$[\, f \mid x_1 \colon y_1 \mid \ldots \mid x_n \colon y_n \,]$$

for the function whose domain is the union of the domain of f with $\{x_1, \ldots, x_n\}$, that maps each x_i into y_i and all other members x of the domain of f into $f\, x$.

—

For heaps, we write

$$h_0 \perp h_1$$

when h_0 and h_1 have disjoint domains, and

$$h_0 \cdot h_1$$

to denote the union of heaps with disjoint domains.

—

Free Variables

For any phrase p,

$\mathsf{FV}(p)$ denotes the set of variables occurring free in p.

There are no binding constructions in expressions or boolean expressions, so that for these phrases $\mathsf{FV}(e)$ is the set of all variables occurring in e. In assertions, quantifiers are binding constructions. In commands, declarations will be binding constructions.

—

The scope of a binding construction is the phrase immediately following the binding occurrence of a variable, except in

$$\mathbf{newvar}\ v = \underline{e}\ \mathbf{in}\ c$$

$$\bigodot_{v=\underline{e_0}}^{\underline{e_1}}\ p,$$

where the underline phrases are excluded from the scope.

—

Total Substitution

For any phrase p such that $\mathsf{FV}(p) \subseteq \{v_1, \ldots, v_n\}$, we write

$$p/v_1 \to e_1, \ldots, v_n \to e_n$$

to denote the phrase obtained from p by simultaneously substituting each expression e_i for the variable v_i, (When there are bound variables in p, they will be renamed to avoid capture.)

—

The Total Substitution Law for Expressions

Proposition 1 *Let δ abbreviate the substitution*

$$v_1 \to e_1, \ldots, v_n \to e_n,$$

let s be a store such that $\mathsf{FV}(e_1) \cup \cdots \cup \mathsf{FV}(e_n) \subseteq \mathsf{dom}\, s$, and let

$$\widehat{s} = [\, v_1 \colon [\![e_1]\!]_{\mathsf{exp}} s \mid \ldots \mid v_n \colon [\![e_n]\!]_{\mathsf{exp}} s \,].$$

If e is an expression (or boolean expression) such that $\mathsf{FV}(e) \subseteq \{v_1, \ldots, v_n\}$, then

$$[\![e/\delta]\!]_{\mathsf{exp}} s = [\![e]\!]_{\mathsf{exp}} \widehat{s}.$$

—

Partial Substitution

When $\mathsf{FV}(p)$ is not a subset of $\{v_1, \ldots, v_n\}$,

$$p/v_1 \to e_1, \ldots, v_n \to e_n$$

abbreviates

$$p/v_1 \to e_1, \ldots, v_n \to e_n, v'_1 \to v'_1, \ldots, v'_k \to v'_k,$$

where $\{v'_1, \ldots, v'_k\} = \mathsf{FV}(p) - \{v_1, \ldots, v_n\}$.

—

The Partial Substitution Law for Expressions

Proposition 2 *Suppose e is an expression (or boolean expression), and let δ abbreviate the substitution*

$$v_1 \to e_1, \ldots, v_n \to e_n,$$

Then let s be a store such that

$$(\mathsf{FV}(e) - \{v_1, \ldots, v_n\}) \cup \mathsf{FV}(e_1) \cup \cdots \cup \mathsf{FV}(e_n) \subseteq \mathsf{dom}\, s,$$

and let

$$\widehat{s} = [\, s \mid v_1 \colon [\![e_1]\!]_{\mathsf{exp}} s \mid \ldots \mid v_n \colon [\![e_n]\!]_{\mathsf{exp}} s \,].$$

Then

$$[\![e/\delta]\!]_{\mathsf{exp}} s = [\![e]\!]_{\mathsf{exp}} \widehat{s}.$$

—

The Meaning of Assertions

When s is a store, h is a heap, and p is an assertion whose free variables belong to the domain of s, we write

$$s, h \models p$$

to indicate that the state s, h *satisfies* p, or p *is true in* s, h, or p *holds in* s, h. Then:

$$\begin{aligned}
s, h \models b \;&\text{ iff }\; [\![b]\!]_{\mathsf{boolexp}} s = \mathbf{true},\\
s, h \models \neg\, p \;&\text{ iff }\; s, h \models p \text{ is false},\\
s, h \models p_0 \wedge p_1 \;&\text{ iff }\; s, h \models p_0 \text{ and } s, h \models p_1\\
&\qquad\qquad \text{(and similarly for } \vee,\ \Rightarrow,\ \Leftrightarrow),\\
s, h \models \forall v.\ p \;&\text{ iff }\; \forall x \in \mathbf{Z}.\ [\, s \mid v\colon x \,], h \models p,\\
s, h \models \exists v.\ p \;&\text{ iff }\; \exists x \in \mathbf{Z}.\ [\, s \mid v\colon x \,], h \models p,\\
s, h \models \mathbf{emp} \;&\text{ iff }\; \mathsf{dom}\, h = \{\},\\
s, h \models e \mapsto e' \;&\text{ iff }\; \mathsf{dom}\, h = \{[\![e]\!]_{\mathsf{exp}} s\} \text{ and}\\
&\qquad\qquad h([\![e]\!]_{\mathsf{exp}} s) = [\![e']\!]_{\mathsf{exp}} s,\\
s, h \models p_0 \ * \ p_1 \;&\text{ iff }\; \exists h_0, h_1.\ h_0 \perp h_1 \text{ and } h_0 \cdot h_1 = h \text{ and}\\
&\qquad\qquad s, h_0 \models p_0 \text{ and } s, h_1 \models p_1,\\
s, h \models p_0 \mathrel{-\!\!*} p_1 \;&\text{ iff }\; \forall h'.\ (h' \perp h \text{ and } s, h' \models p_0) \text{ implies}\\
&\qquad\qquad s, h \cdot h' \models p_1.
\end{aligned}$$

When $s, h \models p$ holds for all states s, h (such that the domain of s contains the free variables of p), we say that p is *valid*.

When $s, h \models p$ holds for some state s, h, we say that p is *satisfiable*.

—

For Instance

$$s, h \vDash \mathsf{x} \mapsto 0 \ * \ \mathsf{y} \mapsto 1$$

iff $\exists h_0, h_1.\ h_0 \perp h_1$ and $h_0 \cdot h_1 = h$
and $s, h_0 \vDash \mathsf{x} \mapsto 0$
and $s, h_1 \vDash \mathsf{y} \mapsto 1$

iff $\exists h_0, h_1.\ h_0 \perp h_1$ and $h_0 \cdot h_1 = h$
and $\mathsf{dom}\, h_0 = \{s\,\mathsf{x}\}$ and $h_0(s\,\mathsf{x}) = 0$
and $\mathsf{dom}\, h_1 = \{s\,\mathsf{y}\}$ and $h_1(s\,\mathsf{y}) = 1$

iff $s\,\mathsf{x} \neq s\,\mathsf{y}$
and $\mathsf{dom}\, h = \{s\,\mathsf{x}, s\,\mathsf{y}\}$
and $h(s\,\mathsf{x}) = 0$ and $h(s\,\mathsf{y}) = 1$

iff $s\,\mathsf{x} \neq s\,\mathsf{y}$ and $h = [\,s\,\mathsf{x}\colon 0 \mid s\,\mathsf{y}\colon 1\,]$.

—

Examples

$$
\begin{aligned}
s, h \vDash \mathsf{x} \mapsto \mathsf{y} &\text{ iff } \operatorname{dom} h = \{s\,\mathsf{x}\} \text{ and } h(s\,\mathsf{x}) = s\,\mathsf{y} \\
s, h \vDash \mathsf{x} \mapsto - &\text{ iff } \operatorname{dom} h = \{s\,\mathsf{x}\} \\
s, h \vDash \mathsf{x} \hookrightarrow \mathsf{y} &\text{ iff } s\,\mathsf{x} \in \operatorname{dom} h \text{ and } h(s\,\mathsf{x}) = s\,\mathsf{y} \\
s, h \vDash \mathsf{x} \hookrightarrow - &\text{ iff } s\,\mathsf{x} \in \operatorname{dom} h \\
s, h \vDash \mathsf{x} \mapsto \mathsf{y}, \mathsf{z} &\text{ iff } h = [\,s\,\mathsf{x} \colon s\,\mathsf{y} \mid s\,\mathsf{x} + 1 \colon s\,\mathsf{z}\,] \\
s, h \vDash \mathsf{x} \mapsto -, - &\text{ iff } \operatorname{dom} h = \{s\,\mathsf{x}, s\,\mathsf{x} + 1\} \\
s, h \vDash \mathsf{x} \hookrightarrow \mathsf{y}, \mathsf{z} &\text{ iff } h \supseteq [\,s\,\mathsf{x} \colon s\,\mathsf{y} \mid s\,\mathsf{x} + 1 \colon s\,\mathsf{z}\,] \\
s, h \vDash \mathsf{x} \hookrightarrow -, - &\text{ iff } \operatorname{dom} h \supseteq \{s\,\mathsf{x}, s\,\mathsf{x} + 1\}.
\end{aligned}
$$

—

More Examples of $*$

Suppose s x and s y are distinct addresses, so that

$$h_0 = [\, s\,\mathsf{x}\colon 0\,] \qquad \text{and} \qquad h_1 = [\, s\,\mathsf{y}\colon 1\,]$$

are heaps with disjoint domains. Then

If p is:	then $s, h \vDash p$ iff:
$\mathsf{x} \mapsto 0$	$h = h_0$
$\mathsf{y} \mapsto 1$	$h = h_1$
$\mathsf{x} \mapsto 0 \ * \ \mathsf{y} \mapsto 1$	$h = h_0 \cdot h_1$
$\mathsf{x} \mapsto 0 \ * \ \mathsf{x} \mapsto 0$	$\mathbf{false}$
$\mathsf{x} \mapsto 0 \vee \mathsf{y} \mapsto 1$	$h = h_0$ or $h = h_1$
$\mathsf{x} \mapsto 0 \ * \ (\mathsf{x} \mapsto 0 \vee \mathsf{y} \mapsto 1)$	$h = h_0 \cdot h_1$
$(\mathsf{x} \mapsto 0 \vee \mathsf{y} \mapsto 1) \ * \ (\mathsf{x} \mapsto 0 \vee \mathsf{y} \mapsto 1)$	$h = h_0 \cdot h_1$
$\mathsf{x} \mapsto 0 \ * \ \mathsf{y} \mapsto 1 \ * \ (\mathsf{x} \mapsto 0 \vee \mathsf{y} \mapsto 1)$	$\mathbf{false}$
$\mathsf{x} \mapsto 0 \ * \ \mathbf{true}$	$h_0 \subseteq h$
$\mathsf{x} \mapsto 0 \ * \ \neg\, \mathsf{x} \mapsto 0$	$h_0 \subseteq h.$

—

The Partial Substitution Law for Assertions

Proposition 3 *Suppose p is an assertion, and let δ abbreviate the substitution*

$$v_1 \to e_1, \ldots, v_n \to e_n,$$

Then let s be a store such that $(\mathsf{FV}(p) - \{v_1, \ldots, v_n\}) \cup \mathsf{FV}(e_1) \cup \cdots \cup \mathsf{FV}(e_n) \subseteq \mathsf{dom}\, s$*, and let*

$$\widehat{s} = [\, s \mid v_1 \colon [\![e_1]\!]_{\mathsf{exp}} s \mid \ldots \mid v_n \colon [\![e_n]\!]_{\mathsf{exp}} s \,].$$

Then

$$s, h \vDash (p/\delta) \text{ iff } \widehat{s}, h \vDash p.$$

—

Inference Rules

$$\frac{\mathcal{P}_1 \qquad \cdots \qquad \mathcal{P}_n}{\mathcal{C}}$$

(zero or more premisses)
(one conclusion)

Inference

Inference Rules

$$\frac{p_0 \qquad p_0 \Rightarrow p_1}{p_1}$$

$$e_2 = e_1 \Rightarrow e_1 = e_2$$

$$\mathrm{x} + 0 = \mathrm{x}$$

Instances

$$\frac{\mathrm{x} + 0 = \mathrm{x} \qquad \mathrm{x} + 0 = \mathrm{x} \Rightarrow \mathrm{x} = \mathrm{x} + 0}{\mathrm{x} = \mathrm{x} + 0}$$

$$\mathrm{x} + 0 = \mathrm{x} \Rightarrow \mathrm{x} = \mathrm{x} + 0$$

$$\mathrm{x} + 0 = \mathrm{x}$$

A Proof

$$\mathrm{x} + 0 = \mathrm{x}$$
$$\mathrm{x} + 0 = \mathrm{x} \Rightarrow \mathrm{x} = \mathrm{x} + 0$$
$$\mathrm{x} = \mathrm{x} + 0.$$

—

Notice:

- Metavariables are in italics (or Greek), object variables are in sans serif.
- An inference rule is *sound* iff, for every instance, if the premisses are all valid, then the conclusion is valid.
- An *axiom schema* is an inference rule with zero premisses.
- An *axiom* is an axiom schema with no metavariables.

—

A Subtlety

$\dfrac{p}{q}$ is sound iff, for all instances,
if p is valid, then q is valid, i.e.,
if p holds in all states, then q holds in all states.

$\dfrac{}{p \Rightarrow q}$ is sound iff, for all instances,
$p \Rightarrow q$ is valid, i.e.,
for all states, if p holds, then q holds.

For example,

$$\frac{p}{\forall v.\ p} \quad \text{e.g.} \quad \frac{\mathsf{x} + \mathsf{y} = \mathsf{y} + \mathsf{x}}{\forall \mathsf{x}.\ \mathsf{x} + \mathsf{y} = \mathsf{y} + \mathsf{x}} \quad \text{or} \quad \frac{\mathsf{x} = 0}{\forall \mathsf{x}.\ \mathsf{x} = 0}$$

is sound, but

$$\frac{}{p \Rightarrow \forall v.\ p} \quad \text{e.g.} \quad \frac{}{\mathsf{x} = 0 \Rightarrow \forall \mathsf{x}.\ \mathsf{x} = 0}$$

is not sound.

—

Inference Rules for Predicate Logic

$$\frac{p \qquad p \Rightarrow q}{q} \quad \text{(modus ponens)}$$

$$\frac{p \Rightarrow q}{p \Rightarrow (\forall v.\ q)} \quad \text{when } v \notin \mathsf{FV}(p)$$

$$\frac{p \Rightarrow q}{(\exists v.\ p) \Rightarrow q} \quad \text{when } v \notin \mathsf{FV}(q).$$

Axiom Schema

$$p \Rightarrow (q \Rightarrow p)$$
$$(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$$
$$(p \wedge q) \Rightarrow p$$
$$(p \wedge q) \Rightarrow q$$
$$p \Rightarrow (q \Rightarrow (p \wedge q))$$
$$p \Rightarrow (p \vee q)$$
$$q \Rightarrow (p \vee q)$$
$$(p \Rightarrow r) \Rightarrow ((q \Rightarrow r) \Rightarrow ((p \vee q) \Rightarrow r))$$
$$(p \Rightarrow q) \Rightarrow ((p \Rightarrow \neg q) \Rightarrow \neg p)$$
$$\neg(\neg p) \Rightarrow p$$
$$(p \Leftrightarrow q) \Rightarrow ((p \Rightarrow q) \wedge (q \Rightarrow p))$$
$$((p \Rightarrow q) \wedge (q \Rightarrow p)) \Rightarrow (p \Leftrightarrow q)$$
$$(\forall v.\ p) \Rightarrow (p/v \rightarrow e)$$
$$(p/v \rightarrow e) \Rightarrow (\exists v.\ p).$$

—

Inference Rules for $*$ and $-\!\!*$

$$p_0 * p_1 \Leftrightarrow p_1 * p_0$$

$$(p_0 * p_1) * p_2 \Leftrightarrow p_0 * (p_1 * p_2)$$

$$p * \mathbf{emp} \Leftrightarrow p$$

$$(p_0 \vee p_1) * q \Leftrightarrow (p_0 * q) \vee (p_1 * q)$$

$$(p_0 \wedge p_1) * q \Rightarrow (p_0 * q) \wedge (p_1 * q)$$

$$(\exists x.\, p_0) * p_1 \Leftrightarrow \exists x.\, (p_0 * p_1) \quad \text{when } x \text{ not free in } p_1$$

$$(\forall x.\, p_0) * p_1 \Rightarrow \forall x.\, (p_0 * p_1) \quad \text{when } x \text{ not free in } p_1$$

$$\frac{p_0 \Rightarrow p_1 \qquad q_0 \Rightarrow q_1}{p_0 * q_0 \Rightarrow p_1 * q_1} \quad \text{(monotonicity)}$$

$$\frac{p_0 * p_1 \Rightarrow p_2}{p_0 \Rightarrow (p_1 -\!\!* p_2)} \quad \text{(currying)} \qquad \frac{p_0 \Rightarrow (p_1 -\!\!* p_2)}{p_0 * p_1 \Rightarrow p_2.} \quad \text{(decurrying)}$$

Some Axiom Schemata for $\mapsto$ and $\hookrightarrow$

$$e_0 \mapsto e_0' \wedge e_1 \mapsto e_1' \Leftrightarrow e_0 \mapsto e_0' \wedge e_0 = e_1 \wedge e_0' = e_1'$$

$$e_0 \hookrightarrow e_0' \ * \ e_1 \hookrightarrow e_1' \Rightarrow e_0 \neq e_1$$

$$\mathbf{emp} \Leftrightarrow \forall x.\ \neg(x \hookrightarrow -)$$

$$(e \hookrightarrow e') \wedge p \Rightarrow (e \mapsto e') \ * \ ((e \mapsto e') \mathrel{-\!\!*} p).$$

—

Pure Assertions

An assertion p is *pure* iff, for all stores s and all heaps h and h',

$$s, h \vDash p \text{ iff } s, h' \vDash p.$$

A sufficient syntactic criteria is that an assertion is pure if it does not contain $\mathbf{emp}$, $\mapsto$, or $\hookrightarrow$.

—

Axiom Schemata for Purity

$p_0 \wedge p_1 \Rightarrow p_0 \ * \ p_1$	when p_0 or p_1 is pure
$p_0 \ * \ p_1 \Rightarrow p_0 \wedge p_1$	when p_0 and p_1 are pure
$(p \wedge q) \ * \ r \Leftrightarrow (p \ * \ r) \wedge q$	when q is pure
$(p_0 \mathrel{-\!\!*} p_1) \Rightarrow (p_0 \Rightarrow p_1)$	when p_0 is pure
$(p_0 \Rightarrow p_1) \Rightarrow (p_0 \mathrel{-\!\!*} p_1)$	when p_0 and p_1 are pure.

—

Strictly Exact Assertions (Yang)

An assertion is *strictly exact* iff, for all stores s and all heaps h and h',

$$s, h \vDash p \text{ and } s, h' \vDash p \text{ implies } h = h'.$$

Examples of Strictly Exact Assertions

- $\mathbf{emp}$.
- $e \mapsto e'$.
- $p \ * \ q$, when p and q are strictly exact.
- $p \wedge q$, when p or q is strictly exact.
- p, when $p \Rightarrow q$ is valid and q is strictly exact.

—

Proposition 4 *When q is strictly exact,*

$$((q \ * \ \mathbf{true}) \wedge p) \Rightarrow (q \ * \ (q \mathrel{-\!\!*} p))$$

is valid.

PROOF Suppose $s, h \vDash (q \ * \ \mathbf{true}) \wedge p$, so that $s, h \vDash q \ * \ \mathbf{true}$ and $s, h \vDash p$. Then there are heaps h_0 and h_1 such that $h_0 \perp h_1$, $h_0 \cdot h_1 = h$, and $s, h_0 \vDash q$.

To see that $s, h_1 \vDash q \mathrel{-\!\!*} p$, let h' be any heap such that $h' \perp h_1$ and $s, h' \vDash q$. Since q is strictly exact, $h' = h_0$, so that $h' \cdot h_1 = h_0 \cdot h_1 = h$, and thus $s, h' \cdot h_1 \vDash p$.

Then $s, h_0 \cdot h_1 \vDash q \ * \ (q \mathrel{-\!\!*} p)$, so that $s, h \vDash q \ * \ (q \mathrel{-\!\!*} p)$.

END OF PROOF

For example, taking q to be the strictly exact assertion $e \mapsto e'$ gives the final axiom schema for $\mapsto$:

$$(e \hookrightarrow e') \wedge p \Rightarrow (e \mapsto e') \ * \ ((e \mapsto e') \mathrel{-\!\!*} p).$$

—

Precise Assertions

An assertion q is *precise* iff

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

$$s, h' \models q.$$

—

Examples of Precise Assertions

- Strictly exact assertions.
- $e \mapsto -$.
- $p \ * \ q$, when p and q are precise.
- $p \wedge q$, when p or q is precise.
- p, when $p \Rightarrow q$ is valid and q is precise.
- list $\alpha \ e$ and $\exists \alpha$. list $\alpha \ e$.
- tree $\tau \ (e)$ and $\exists \tau$. tree $\tau \ (e)$.

—

Examples of Imprecise Assertions

- $\mathbf{true}$
- $\mathbf{emp} \vee \mathsf{x} \mapsto 10$
- $\mathsf{x} \mapsto 10 \vee \mathsf{y} \mapsto 10$
- $\exists \mathsf{x}.\ \mathsf{x} \mapsto 10$
- $\mathsf{dag}\ \tau\ (\mathsf{i})$
- $\exists \tau.\ \mathsf{dag}\ \tau\ (\mathsf{i})$

—

Preciseness and Distributivity

The semi-distributive laws

$$(p_0 \wedge p_1) \ * \ q \Rightarrow (p_0 \ * \ q) \wedge (p_1 \ * \ q)$$
$$(\forall x.\ p) \ * \ q \Rightarrow \forall x.\ (p \ * \ q) \quad \text{when } x \text{ not free in } q$$

are valid for all assertions. But their converses

$$(p_0 \ * \ q) \wedge (p_1 \ * \ q) \Rightarrow (p_0 \wedge p_1) \ * \ q$$
$$\forall x.\ (p \ * \ q) \Rightarrow (\forall x.\ p) \ * \ q \quad \text{when } x \text{ not free in } q$$

are not. For example, when

$$s(\mathsf{x}) = 1 \quad s(\mathsf{y}) = 2 \quad h = [\,1\colon 10 \mid 2\colon 20\,],$$

the assertion

$$(\mathsf{x} \mapsto 10 * (\mathsf{x} \mapsto 10 \vee \mathsf{y} \mapsto 20)) \wedge (\mathsf{y} \mapsto 20 * (\mathsf{x} \mapsto 10 \vee \mathsf{y} \mapsto 20))$$

is true, but

$$((\mathsf{x} \mapsto 10 \wedge \mathsf{y} \mapsto 20) \ * \ (\mathsf{x} \mapsto 10 \vee \mathsf{y} \mapsto 20))$$

is false.

However, the converses are valid when q is precise.

—

Preciseness and Distributivity (continued)

Proposition 5 *When q is precise,*

$$(p_0 \ * \ q) \wedge (p_1 \ * \ q) \Rightarrow (p_0 \wedge p_1) \ * \ q$$

is valid. When q is precise and x is not free in q,

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

is valid.

PROOF (first law) Suppose $s, h \vDash (p_0 \ * \ q) \wedge (p_1 \ * \ q)$. Then there are:

- An $h_0 \subseteq h$ such that $s, h - h_0 \vDash p_0$ and $s, h_0 \vDash q$, and
- An $h_1 \subseteq h$ such that $s, h - h_1 \vDash p_1$ and $s, h_1 \vDash q$.

Thus, since q is precise,

$$h_0 = h_1$$
$$h - h_0 = h - h_1$$
$$s, h - h_0 \vDash p_0 \wedge p_1$$
$$s, h \vDash (p_0 \wedge p_1) \ * \ q.$$

END OF PROOF

—

Intuitionistic Assertions

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

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

Assume i and i' are intuitionistic assertions, and p is any assertion. Then:

- The following assertions are intuitionistic:

Any pure assertion	$p * i$
$p \mathrel{-\!\!*} i$	$i \mathrel{-\!\!*} p$
$i \wedge i'$	$i \vee i'$
$\forall v.\ i$	$\exists v.\ i$
$\mathsf{dag}\ \tau\ (e)$	$\exists \tau.\ \mathsf{dag}\ \tau\ (e)$,

  and as special cases:

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

—

- The following inference rules are sound:

$$(i \ * \ i') \Rightarrow (i \wedge i')$$

$$(i \ * \ p) \Rightarrow i \qquad i \Rightarrow (p \mathrel{-\!\!*} i)$$

$$\frac{p \Rightarrow i}{(p \ * \ \mathbf{true}) \Rightarrow i} \qquad \frac{i \Rightarrow p}{i \Rightarrow (\mathbf{true} \mathrel{-\!\!*} p).}$$

The last two of these rules, in conjunction with the rules

$$p \Rightarrow (p \ * \ \mathbf{true}) \qquad (\mathbf{true} \mathrel{-\!\!*} p) \Rightarrow p,$$

which hold for all assertions, imply that

- $p * \mathbf{true}$ is the strongest intuitionistic assertion weaker than p.
- $\mathbf{true} \mathrel{-\!\!*} p$ is the weakest intuitionistic assertion that is stronger than p.
- $i \Leftrightarrow (i \ * \ \mathbf{true})$.
- $(\mathbf{true} \mathrel{-\!\!*} i) \Leftrightarrow i$.

—

The Intuitionistic Version of Separation Logic

If we define the operations

$$\begin{aligned} \overset{\mathsf{i}}{\neg}\, p &\overset{\mathsf{def}}{=} \mathbf{true} \mathrel{-\!\!*} (\neg\, p) \\ p \overset{\mathsf{i}}{\Rightarrow} q &\overset{\mathsf{def}}{=} \mathbf{true} \mathrel{-\!\!*} (p \Rightarrow q) \\ p \overset{\mathsf{i}}{\Leftrightarrow} q &\overset{\mathsf{def}}{=} \mathbf{true} \mathrel{-\!\!*} (p \Leftrightarrow q), \end{aligned}$$

then the assertions built from pure assertions and $e \hookrightarrow e'$, using these operations and $\wedge$, $\vee$, $\forall$, $\exists$, $*$, and $-\!\!*$ form the intuitionistic version of separation logic.

—

Supported Assertions

It is easily seen that no assertion that is satisfiable (i.e. that holds in some state) can be both precise and intuitionistic.

So what, in the intuitionistic world, replaces the concept of "precise"?

An assertion q is *supported* iff, for all s, h_0, and h_1,

- if $h_0 \cup h_1$ is a function, and $s, h_0 \models q$ and $s, h_1 \models q$ are true,
- then there is an h' such that $h' \subseteq h_0$, $h' \subseteq h_1$, and $s, h' \models q$ is true.

Equivalently,

Proposition 6 *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.

—

Proposition 6 *An assertion q is* supported *iff, for all s and h, if the set*

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

is nonempty, then it has a least element.

PROOF Suppose that q is supported, fix s and h, and let h_0 be a member of H with minimum domain size, and h_1 be any member of H. Since h_0 and h_1 are both subsets of h, $h_0 \cup h_1$ must be a function. Then the first definition guarantees that there is an $h' \in H$ that is a subset of both h_0 and h_1. But h' must be equal to h_0, since otherwise it would have a smaller domain size. Thus $h_0 \subseteq h_1$ for every $h_1 \in H$.

On the other hand, suppose that q meets the conditions of the proposition, and $h_0 \cup h_1$ is a function, $s, h_0 \vDash q$ and $s, h_1 \vDash q$ are true. Take h to be $h_0 \cup h_1$, so that $h_0, h_1 \in H$. Then take h' to be the least element of H. END OF PROOF

—

Examples

- Imprecise, Intuitionistic, and Supported

$$\mathbf{true} \qquad \mathsf{x} \hookrightarrow 10 \qquad \mathsf{x} \hookrightarrow 10 \wedge \mathsf{y} \hookrightarrow 10$$
$$\mathsf{dag}\ \tau\ (\mathsf{i}) \qquad \exists \tau.\ \mathsf{dag}\ \tau\ (\mathsf{i})$$

- Imprecise, Intuitionistic, Unsupported

$$\mathsf{x} \hookrightarrow 10 \vee \mathsf{y} \hookrightarrow 10 \qquad \exists \mathsf{x}.\ \mathsf{x} \hookrightarrow 10 \qquad \neg\, \mathbf{emp}$$

- Imprecise, Nonintuitionistic, Supported

$$\mathbf{emp} \vee \mathsf{x} \mapsto 10$$

- Imprecise, Nonintuitionistic, Unsupported

$$\mathsf{x} \mapsto 10 \vee \mathsf{y} \mapsto 10 \qquad \exists \mathsf{x}.\ \mathsf{x} \mapsto 10$$

—

Supported Assertions and Distributivity

Proposition 7 *When p_0 and p_1 are intuitionistic and q is supported,*

$$(p_0 \ * \ q) \wedge (p_1 \ * \ q) \Rightarrow (p_0 \wedge p_1) \ * \ q$$

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

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

is valid.

PROOF (of the first law): Suppose $s, h \vDash (p_0 \ * \ q) \wedge (p_1 \ * \ q)$. Then there are:

An $h_0 \subseteq h$ such that $s, h - h_0 \vDash p_0$ and $s, h_0 \vDash q$,

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

Then, since q is supported and $h_0 \cup h_1$ is a function, there is an $h' \subseteq h_0, h_1$ such that $s, h' \vDash q$. Moreover, since $h - h_0, h - h_1 \subseteq h - h'$, and p_0 and p_1 are intuitionistic,

$$s, h - h' \vDash p_0 \wedge p_1,$$

and therefore

$$s, h \vDash (p_0 \wedge p_1) \ * \ q.$$

END OF PROOF

—

The Operator $- * \mathbf{true}$

We have already seen that, if p is any assertion, then $p * \mathbf{true}$ is intuitionistic, and if i is intuitionistic, then $i \Leftrightarrow (i * \mathbf{true})$.

Thus, $- * \mathbf{true}$ maps arbitrary assertions into intuitionistic assertions, and acts as an identity (up to equivalence of assertions) on the latter.

Moreover, $- * \mathbf{true}$ maps precise assertions into supported intuitionistic assertions and acts as an identity (up to equivalence of assertions) on the latter. (This is a consequence of the following proposition.)

—

Proposition 8 *(1) If p is precise, then p is supported.*
(2) q is supported iff $q * \mathbf{true}$ is supported.

PROOF (1) If p is precise, then, for any s and h, the set

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

contains at most one element.

(2) Suppose q is supported, $h_0 \cup h_1$ is a function, $s, h_0 \models q * \mathbf{true}$ and $s, h_1 \models q * \mathbf{true}$. Then there are $h'_0 \subseteq h_0$ and $h'_1 \subseteq h_1$ such that $s, h'_0 \models q$ and $s, h'_1 \models q$, and since q is supported, there is an h' that is a subset of h'_0 and h'_1, and therefore h_0 and h_1, such that $s, h' \models q$, and therefore $s, h' \models q * \mathbf{true}$.

Suppose $q * \mathbf{true}$ is supported, $h_0 \cup h_1$ is a function, $s, h_0 \models q$ and $s, h_1 \models q$. Then $s, h_0 \models q * \mathbf{true}$ and $s, h_1 \models q * \mathbf{true}$, and since $q * \mathbf{true}$ is supported, there is a common subset h' of h_0 and h_1 such that $s, h' \models q * \mathbf{true}$. But then there is a subset h'' of h', and therefore of h_0 and h_1, such that $s, h'' \models q$.

END OF PROOF

—

The Precising Operation

We define the *precising* operation:

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

For example,

$$\mathsf{Pr}\ \mathbf{true} \text{ iff } \mathbf{emp}$$

$$\mathsf{Pr}\ (\mathsf{x} \hookrightarrow 10) \text{ iff } \mathsf{x} \mapsto 10$$

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

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

The operation Pr maps supported operations into precise operations, and acts as an identity on the latter. (This is a consequence of the following proposition.)

—

Proposition 9 *(1) If p is supported, then* Pr p *is precise.*
(2) If p is precise, then Pr $p \Leftrightarrow p$.

PROOF (1) Suppose p is supported, and $h_0, h_1 \subseteq h$ are such that $s, h_0 \models \mathsf{Pr}\ p$ and $s, h_1 \models \mathsf{Pr}\ p$.

We must show $h_0 = h_1$. Assume the contrary. Since $s, h_0 \models p$, $s, h_1 \models p$, and p is supported, there is a common subset h' of h_0 and h_1 such that $s, h' \models p$. Since $h_0 \neq h_1$, however, h' must be a proper subset of h_i for $i = 0$ or $i = 1$. Thus $h_i = h' \cdot (h_i - h')$, where $s, h_i - h' \models \neg\,\mathbf{emp}$. Then $s, h_i \models p * \neg\,\mathbf{emp}$, which contradicts $s, h_i \models \mathsf{Pr}\ p$.

(2) Obviously, Pr $p \Rightarrow p$. To show the opposite implication when p is precise, assume $s, h \models p$. If $s, h \models p * \neg\,\mathbf{emp}$ held, then there would be a proper subset h' of h such that $s, h' \models p$, which would contradict the preciseness of p. Thus $s, h \models \neg(p * \neg\,\mathbf{emp})$.

END OF PROOF

—

Relating $- * \mathbf{true}$ and Pr

Proposition 10 *(1)* $\mathsf{Pr}\ (p * \mathbf{true}) \Rightarrow p$.

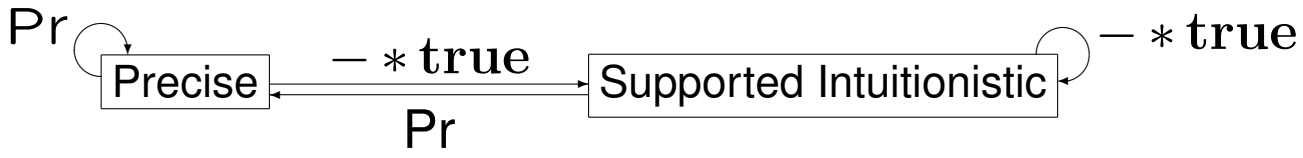
(2) $p \Rightarrow \mathsf{Pr}\ (p * \mathbf{true})$ *when* p *is precise.*

(3) $(\mathsf{Pr}\ q) * \mathbf{true} \Rightarrow q$ *when* q *is intuitionistic.*

(4) $q \Rightarrow (\mathsf{Pr}\ q) * \mathbf{true}$ *when* q *is supported.*

Therefore, $\mathsf{Pr}\ (p * \mathbf{true}) \Leftrightarrow p$ *when* p *is precise, and* $(\mathsf{Pr}\ q) * \mathbf{true} \Rightarrow q$ *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, and act as identities on these sets:



—

PROOF

(1) Pr $(p * \mathbf{true}) \Rightarrow p$.

Suppose $s, h \vDash$ Pr $(p * \mathbf{true})$. Then $s, h \vDash p * \mathbf{true}$ and $s, h \vDash \neg(p * \mathbf{true} * \neg\mathbf{emp})$, and there is an $h' \subseteq h$ such that $s, h' \vDash p$. If $h' = h$, we are done; otherwise, h' is a proper subset of h, so that $s, h \vDash p * \mathbf{true} * \neg\mathbf{emp}$, which contradicts $s, h \vDash \neg(p * \mathbf{true} * \neg\mathbf{emp})$.

(2) $p \Rightarrow$ Pr $(p * \mathbf{true})$ when p is precise.

Suppose $s, h \vDash p$. Then $s, h \vDash p * \mathbf{true}$. Moreover, $s, h \vDash \neg(p * \mathbf{true} * \neg\mathbf{emp})$, for otherwise $s, h \vDash p * \mathbf{true} * \neg\mathbf{emp}$ would imply that there is a proper subset h' of h such that $s, h' \vDash p$, which would contradict the preciseness of p.

(3) (Pr q) $* \mathbf{true} \Rightarrow q$ when q is intuitionistic.

Suppose $s, h \vDash$ (Pr q) $* \mathbf{true}$. Then there is an $h' \subseteq h$ such that $s, h' \vDash$ Pr q. Then $s, h' \vDash q$, and since q is intuitionistic, $s, h \vDash q$.

(4) $q \Rightarrow$ (Pr q) $* \mathbf{true}$ when q is supported.

Suppose $s, h \vDash q$. Since q is supported, there is a least $h' \subseteq h$ such that $s, h' \vDash q$. Then $s, h' \vDash$ Pr q, since otherwise $s, h' \vDash q * \neg\mathbf{emp}$, which would imply that a proper subset h'' of h' would satisfy $s, h'' \vDash q$, contradicting the leastness of h'. Thus $s, h \vDash$ (Pr q) $* \mathbf{true}$. END OF PROOF

—

Some Derived Inference Rules

$$\frac{}{q \ * \ (q \mathrel{-\!\!*} p) \Rightarrow p}$$

1. $q \ * \ (q \mathrel{-\!\!*} p) \Rightarrow (q \mathrel{-\!\!*} p) \ * \ q$ $\qquad (p_0 \ * \ p_1 \Rightarrow p_1 \ * \ p_0)$
2. $(q \mathrel{-\!\!*} p) \Rightarrow (q \mathrel{-\!\!*} p)$ $\qquad (p \Rightarrow p)$
3. $(q \mathrel{-\!\!*} p) \ * \ q \Rightarrow p$ $\qquad$ (decurrying, 2)
4. $q \ * \ (q \mathrel{-\!\!*} p) \Rightarrow p$ $\qquad$ (trans impl, 1, 3)

where *transitive implication* is the inference rule

$$\frac{p \Rightarrow q \qquad q \Rightarrow r}{p \Rightarrow r.}$$

—

$$\frac{}{r \Rightarrow (q \mathrel{-\!\!*} (q \ * \ r))}$$

1. $(r \ * \ q) \Rightarrow (q \ * \ r)$ $(p_0 \ * \ p_1 \Rightarrow p_1 \ * \ p_0)$
2. $r \Rightarrow (q \mathrel{-\!\!*} (q \ * \ r))$ (currying, 1)

—

$$\frac{}{(p \ * \ r) \Rightarrow (p \ * \ (q \ -\!\!* \ (q \ * \ r)))}$$

1. $p \Rightarrow p$ ($p \Rightarrow p$)
2. $r \Rightarrow (q \ -\!\!* \ (q \ * \ r))$ (derived above)
3. $(p \ * \ r) \Rightarrow (p \ * \ (q \ -\!\!* \ (q \ * \ r)))$ (monotonicity, 1, 2)

—

$$\frac{p_0 \Rightarrow (q \mathrel{-\!\!*} r) \qquad p_1 \Rightarrow (r \mathrel{-\!\!*} s)}{p_1 \ * \ p_0 \Rightarrow (q \mathrel{-\!\!*} s)}$$

1. $p_1 \Rightarrow p_1$ ($p \Rightarrow p$)
2. $p_0 \Rightarrow (q \mathrel{-\!\!*} r)$ (assumption)
3. $p_0 \ * \ q \Rightarrow r$ (decurrying, 2)
4. $p_1 \ * \ p_0 \ * \ q \Rightarrow p_1 \ * \ r$ (monotonicity, 1, 3)
5. $p_1 \Rightarrow (r \mathrel{-\!\!*} s)$ (assumption)
6. $p_1 \ * \ r \Rightarrow s$ (decurrying, 5)
7. $p_1 \ * \ p_0 \ * \ q \Rightarrow s$ (trans impl, 4, 6)
8. $p_1 \ * \ p_0 \Rightarrow (q \mathrel{-\!\!*} s)$ (currying, 7)

—

$$\frac{p' \Rightarrow p \qquad q \Rightarrow q'}{(p \mathrel{-\!\!*} q) \Rightarrow (p' \mathrel{-\!\!*} q').}$$

1. $(p \mathrel{-\!\!*} q) \Rightarrow (p \mathrel{-\!\!*} q)$ $(p \Rightarrow p)$
2. $p' \Rightarrow p$ (assumption)
3. $(p \mathrel{-\!\!*} q) * p' \Rightarrow (p \mathrel{-\!\!*} q) * p$ (monotonicity, 1, 2)
4. $(p \mathrel{-\!\!*} q) * p \Rightarrow q$ (decurrying, 1)
5. $(p \mathrel{-\!\!*} q) * p' \Rightarrow q$ (trans impl, 3, 4)
6. $q \Rightarrow q'$ (assumption)
7. $(p \mathrel{-\!\!*} q) * p' \Rightarrow q'$ (trans impl, 5, 6)
8. $(p \mathrel{-\!\!*} q) \Rightarrow (p' \mathrel{-\!\!*} q')$ (currying, 7)

—

Exercise 1

Give a formal proof of the valid assertion

$$\big((\mathsf{x} \mapsto \mathsf{y} \ * \ \mathsf{x}' \mapsto \mathsf{y}') \ * \ \mathbf{true}\big) \Rightarrow \big(((\mathsf{x} \mapsto \mathsf{y} \ * \ \mathbf{true}) \wedge (\mathsf{x}' \mapsto \mathsf{y}' \ * \ \mathbf{true})) \wedge \mathsf{x} \neq \mathsf{x}'\big)$$

from the rules in (2.3) and (2.4), and (some of) the following (derived) inference rules for predicate calculus:

$$p \Rightarrow \mathbf{true} \qquad p \Rightarrow p \qquad p \wedge \mathbf{true} \Rightarrow p$$

$$\frac{p \Rightarrow q \qquad q \Rightarrow r}{p \Rightarrow r} \quad \text{(trans impl)}$$

$$\frac{p \Rightarrow q \qquad p \Rightarrow r}{p \Rightarrow q \wedge r} \quad (\wedge\text{-introduction})$$

Your proof will be easier to read if you write it as a sequence of steps rather than a tree. In the inference rules, you should regard $*$ as left associative, e.g.,

$$e_0 \mapsto e_0' \ * \ e_1 \mapsto e_1' \ * \ \mathbf{true} \Rightarrow e_0 \neq e_1$$

stands for

$$(e_0 \mapsto e_0' \ * \ e_1 \mapsto e_1') \ * \ \mathbf{true} \Rightarrow e_0 \neq e_1.$$

For brevity, you may weaken $\Leftrightarrow$ to $\Rightarrow$ when it is the main operator of an axiom. You may also omit instances of the axiom schema $p \Rightarrow p$ when it is used as a premiss of the monotonicity rule.

—

Exercise 2

None of the following axiom schemata are sound. For each, given an instance which is not valid, along with a description of a state in which the instance is false.

$$p_0 * p_1 \Rightarrow p_0 \wedge p_1 \quad \text{(unsound)}$$

$$p_0 \wedge p_1 \Rightarrow p_0 * p_1 \quad \text{(unsound)}$$

$$(p_0 * p_1) \vee q \Rightarrow (p_0 \vee q) * (p_1 \vee q) \quad \text{(unsound)}$$

$$(p_0 \vee q) * (p_1 \vee q) \Rightarrow (p_0 * p_1) \vee q \quad \text{(unsound)}$$

$$(p_0 * q) \wedge (p_1 * q) \Rightarrow (p_0 \wedge p_1) * q \quad \text{(unsound)}$$

$$(p_0 * p_1) \wedge q \Rightarrow (p_0 \wedge q) * (p_1 \wedge q) \quad \text{(unsound)}$$

$$(p_0 \wedge q) * (p_1 \wedge q) \Rightarrow (p_0 * p_1) \wedge q \quad \text{(unsound)}$$

$$\big(\forall x.\ (p_0 * p_1)\big) \Rightarrow\ (\forall x.\ p_0) * p_1 \quad \text{when } x \text{ not free in } p_1 \quad \text{(unsound)}$$

$$(p_0 \Rightarrow p_1) \Rightarrow \big((p_0 * q) \Rightarrow (p_1 * q)\big) \quad \text{(unsound)}$$

$$(p_0 \Rightarrow p_1) \Rightarrow (p_0 \mathrel{-\!\!*} p_1) \quad \text{(unsound)}$$

$$(p_0 \mathrel{-\!\!*} p_1) \Rightarrow (p_0 \Rightarrow p_1) \quad \text{(unsound)}$$

—

Exercise 3

Use the semantics of assertions to show:

a. The soundness of

$$\frac{p_0 \Rightarrow p_1 \qquad q_0 \Rightarrow q_1}{p_0 \ * \ q_0 \Rightarrow p_1 \ * \ q_1} \quad \text{(monotonicity)}$$

$$\frac{p_0 \ * \ p_1 \Rightarrow p_2}{p_0 \Rightarrow (p_1 \mathrel{-\!\!*} p_2)} \quad \text{(currying)} \qquad \frac{p_0 \Rightarrow (p_1 \mathrel{-\!\!*} p_2)}{p_0 \ * \ p_1 \Rightarrow p_2.} \quad \text{(decurrying).}$$

b. When q is pure, the soundness of

$$(p \wedge q) \ * \ r \Leftrightarrow (p \ * \ r) \wedge q.$$

c. When i is intuitionistic, that:

$p \ * \ i, p \mathrel{-\!\!*} i,$ and $i \mathrel{-\!\!*} p$ are intuitionistic.

d. When i is intuitionistic, the soundness of:

$$\frac{p \Rightarrow i}{(p \ * \ \mathbf{true}) \Rightarrow i} \qquad \frac{i \Rightarrow p}{i \Rightarrow (\mathbf{true} \mathrel{-\!\!*} p).}$$

—

Exercise 4

An assertion p is said to be *domain-exact* iff, for all s, h, and h', $s, h \models p$ and $s, h' \models p$ implies $\mathsf{dom}\, h = \mathsf{dom}\, h'$.

Examples of domain-exact assertions include:

- Strictly exact assertions
- $e \mapsto -$
- $p \ * \ q$, when p and q are domain-exact
- $p \wedge q$, when p or q is domain-exact
- p, when $p \Rightarrow q$ is valid and q is domain-exact.

Use the semantics of assertions to show that any domain-exact assertion is precise.

—