Imperial College London

✓ Got 6 ECTS

✗ almost failed exam

✗ 6 Months of Work

Program Logics for Free

Alyssa Renata
alyssa.renata19@imperial.ac.uk

11 February 2025

Domain Theory for Program Reasoning

By Alyssa Renata

(Based on the works of
Samson Abramsky
&
Steve Vickers)

$\underline{\text{def}}$ (WHILE Language)

$A ::= n \in \mathbb{Z} \mid x \in Var \mid A \ \underline{\text{plus}}\ A \mid A \ \underline{\text{times}}\ A \ \ldots$ Arithmetic Expressions

$B ::= \underline{tt} \mid \underline{ff} \mid A \ \underline{\text{eq}}\ A \mid B \ \underline{\text{or}}\ B \mid B \ \underline{\text{and}}\ B \ \ldots$ Boolean Expressions

$C ::= \underline{\text{skip}} \mid \underline{\text{def}}\ x := A \mid C \ \underline{;}\ C \mid \underline{\text{if}}\ B \ \underline{\text{then}}\ C \ \underline{\text{else}}\ C$
$\mid \underline{\text{while}}\ B \ \underline{\text{do}}\ C$

Commands

First Attempt at Semantics

State $\Sigma = Var \longrightarrow \mathbb{Z}$

$[\![C]\!] : \underbrace{\Sigma \longrightarrow \Sigma}_{\text{"State Transformer"}}$

$$\left.\begin{aligned} [\![A]\!] &: \Sigma \longrightarrow \mathbb{Z} \\ [\![B]\!] &: \Sigma \longrightarrow Bool \end{aligned}\right\} \text{Standard Stuff}$$

:)

$[\![\underline{\text{skip}}]\!] = id_\Sigma \qquad [\![C_1 \,;\, C_2]\!] = [\![C_2]\!] \circ [\![C_1]\!]$

$[\![\underline{\text{def}}\ x := A]\!](s) = s[x \mapsto [\![A]\!](s)]$

$$[\![\underline{\text{if}}\ B\ \underline{\text{then}}\ C_1\ \underline{\text{else}}\ C_2]\!](s) = \begin{cases} [\![C_1]\!](s) & \text{if } [\![B]\!](s) = tt \\ [\![C_2]\!](s) & \text{if } [\![B]\!](s) = ff \end{cases}$$

$$[\![\underline{\text{while}}\ B\ \underline{\text{do}}\ C]\!]\ ?$$

$$[\![\text{while } B \text{ do } C]\!](s) = \begin{cases} [\![\text{while } B \text{ do } C]\!] \circ [\![C]\!](s) & \text{if } [\![B]\!](s) = tt \\ s & \text{if } [\![B]\!](s) = ff \end{cases}$$

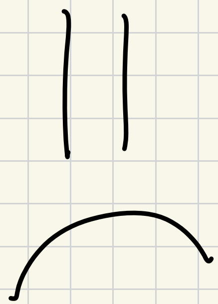
$$[\![\text{while } x > 0 \text{ do skip}]\!]\ ?$$

$$(s(x) > 0)$$

$$[\![\text{while } x>0 \text{ do skip}]\!](s)$$

$$= \begin{cases} [\![\text{—}\,''\,\text{——}]\!] \circ [\![\text{skip}]\!](s) & \text{if } [\![B]\!](s) = tt \\ s & \text{if } [\![B]\!](s) = ff \end{cases}$$

$$= [\![\text{—}\,''\,\text{——}]\!] \circ [\![\text{skip}]\!](s) = [\![\text{—}\,''\,\text{——}]\!](s)$$



Let $F_{B,C} : (\Sigma \to \Sigma) \longrightarrow (\Sigma \to \Sigma)$

$$F_{B,C}(f)(s) = \begin{cases} f \circ [\![C]\!](s) & \text{if } [\![B]\!](s) = T \\ s & \text{if } [\![B]\!](s) = \emptyset \end{cases}$$

Criterion: Solve for f s.t. $f = F_{B,C}(f)$ fixpoint.

Problem: $f = F_{x>0,\,skip}(f) \rightsquigarrow f(s) = \begin{cases} f(s) & \text{if } s(x) > 0 \\ s & \text{if } s(x) \leq 0 \end{cases}$

too many fixpoints.

None are good

Then **best** fixpoint is $f(s) = \begin{cases} \text{undefined} & \text{if } s(x) > 0 \\ s & \text{if } s(x) \leq 0 \end{cases}$

So Change to $[\![C]\!] : \Sigma \longrightarrow \Sigma$.

$f \sqsubseteq g$ iff $\forall s \in \Sigma.\ f(s)\downarrow \implies g(s)\downarrow$ and $f(s) = g(s)$

least fixpoint

Bottom-up Approach to least fixpoint

- $\perp(s) := \text{undefined} \quad \text{for all } s \in \Sigma$

- $$F_{x>0,\, x--}(\perp)(s) = \begin{cases} \perp \circ [\![x--]\!](s) = \text{undef} & s(x) > 0 \\ s & s(x) \leq 0 \end{cases}$$

- $$F(F(\perp))(s) = \begin{cases} F(\perp) \circ [\![x--]\!](s) = \begin{cases} \text{undefined} & s(x) > 1 \\ [\![x--]\!](s) & 0 < s(x) \leq 1 \end{cases} \\ s & s(x) \leq 0 \end{cases}$$

- $$F^3(\perp)(s) = \begin{cases} \text{undefined} & s(x) > 2 \\ [\![x--]\!]^2(s) & s(x) = 2 \\ [\![x--]\!](s) & s(x) = 1 \\ s & s(x) \leq 0 \end{cases}$$

$\vdots$

so $\bot \sqsubseteq F(\bot) \sqsubseteq F^2(\bot) \sqsubseteq \ldots$

when do we hit the complete solution?

$$[\![\text{while } B \text{ do } C]\!] = \text{“} F^{\infty}_{B,C} \text{”}$$

$$= \bigsqcup_{n \in \mathbb{N}} F^{n}_{B,C}(\bot)$$

universal property

$$\boxed{\begin{array}{l} \forall n.\ F^n(\bot) \sqsubseteq \bigsqcup_{n\in\mathbb{N}} F^n(\bot) \\ \text{and whenever } \forall n.\ F^n(\bot) \sqsubseteq g, \text{ then } \bigsqcup_{n\in\mathbb{N}} F^n(\bot) \sqsubseteq g. \end{array}}$$

In $(\Sigma \to \Sigma, \sqsubseteq)$, $\bigsqcup_{n\in\mathbb{N}} F^n(\bot) = \bigcup_{n\in\mathbb{N}} F^n(\bot)$

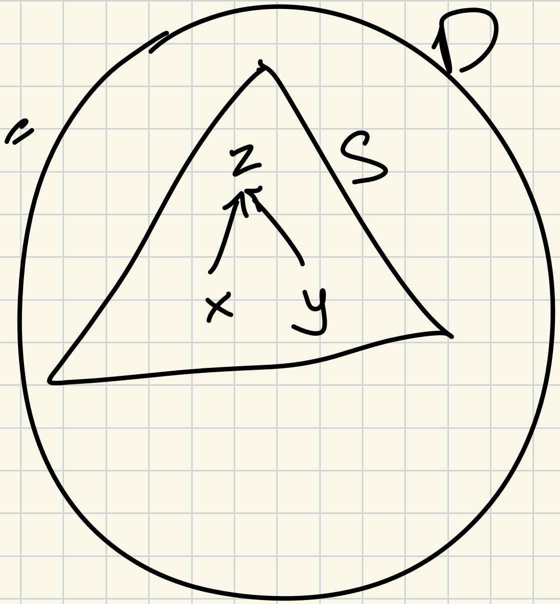
union of graphs

DCPOs

def $(D, \sqsubseteq)$ poset is (pointed) DCPO if

- D has $\perp$ least element $\rightarrow \perp = \bigsqcup \emptyset$
- For every directed $S \subseteq D$, $\bigsqcup S$ exists

"generalized" sequence



$\forall x, y \in S.\ \exists z \in S.\ x \sqsubseteq z \sqsupseteq y.$

NON-example

$\omega = (\mathbb{N}, \leq)$

$\bigsqcup \mathbb{N}$?

Can do

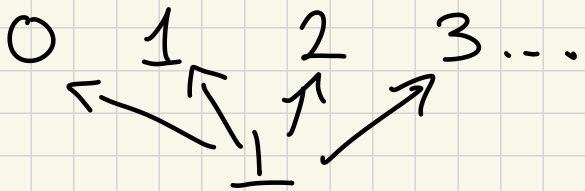
$\omega + 1 = \mathbb{N} \cup \{\mathbb{N}\}$

Examples

- $\Sigma \longrightarrow \Sigma$
- $A \longrightarrow B$

  $A, B \in \mathrm{Set}$

- $\mathbb{N}_\perp = \mathbb{N} \cup \{\perp\}$



- $\Sigma' = \mathbb{N} \rightarrow \mathbb{Z}$

  "Better Σ"

Category of DCPOs

def function $f : D \to E$ is

- **monotone** if $s \sqsubseteq t \Rightarrow f(s) \sqsubseteq f(t)$
- **(Scott) continuous** if monotone and for all (non-empty) directed $S \subseteq D$, $f(\bigsqcup S) = \bigsqcup f[S]$

  $\hookrightarrow$ DO NOT generally want $f(\perp) = \perp$.

theorem (Knaster-Tarski, 19??)

For $F : D \to D$ continuous, $\bigsqcup_{n \in \mathbb{N}} F^n(\perp)$ is least fix point.

$$\perp \sqsubseteq f(\perp) \sqsubseteq f^2(\perp) \sqsubseteq \cdots$$

proof $F(\bigsqcup_{n \in \mathbb{N}} F^n(\perp)) = \bigsqcup_{n \in \mathbb{N}} F^{n+1}(\perp) = \bigsqcup_{n \in \mathbb{N}} F^n(\perp)$.

If $F(d) = d$, then $F^n(\perp) \overset{\text{mono}}{\sqsubseteq} F^n(d) \overset{\text{fix}}{=} d$ so $\bigsqcup_{n \in \mathbb{N}} F^n(\perp) \sqsubseteq d$. $\square$

exercise $F_{B,C}$ is continuous.

Function Space

def

$$[D \to E] \stackrel{\Delta}{=} \{ f : D \to E \mid f \text{ continuous} \}$$

$$f \sqsubseteq_{[D \to E]} g \stackrel{\Delta}{\iff} \forall d \in D.\ f(d) \sqsubseteq_E g(d)$$

$$\bot_{[D \to E]}(d) = \bot_E$$

$$\left(\bigsqcup S \right)(d) = \bigsqcup \{ f(d) \mid f \in S \} \qquad \text{for } S \subseteq [D \to E]$$

Strict Version

$$[D \to_{\bot} E] = \{ f \in [D \to E] \mid f(\bot) = \bot \}$$

example $A \to B \cong [A_\bot \to_\bot B_\bot]$ so $\Sigma = [Var_\bot \to_\bot \mathbb{Z}_\bot]$

Semantics of WHILE

$$[\![A]\!] : [\Sigma \to \mathbb{Z}_\perp]$$

$$[\![n \in \mathbb{Z}]\!](s) = n$$

$$[\![x \in Var]\!](s) = s(x)$$

$$[\![A_1 + A_2]\!](s) = [\![A_1]\!](s) \; [\![+]\!] \; [\![A_2]\!](s)$$

$\vdots$

$$k_1 \; [\![+]\!] \; k_2 = k_1 + k_2$$

$$k_1 \; [\![+]\!] \; \perp = \perp$$

$$\perp \; [\![+]\!] \; k_2 = \perp$$

$$\perp \; [\![+]\!] \; \perp = \perp$$

Semantics of WHILE

$$[\![C]\!] : [\Sigma \to \Sigma]$$

$$[\![\text{skip}]\!] = id_{\Sigma} \qquad [\![C_1 ; C_2]\!] = [\![C_2]\!] \circ [\![C_1]\!]$$

$$[\![\text{def } n := A]\!](s) = \begin{cases} \bot & \text{if } [\![A]\!](s) = \bot \\ s[n := [\![A]\!](s)] & \text{otherwise} \end{cases}$$

$$[\![\text{if } B \text{ then } C_1 \text{ else } C_2]\!](s) = \begin{cases} \bot & \text{if } [\![B]\!](s) = \bot \\ [\![C_1]\!](s) & \text{if } [\![B]\!](s) = tt \\ [\![C_2]\!](s) & \text{if } [\![B]\!](s) = ff \end{cases}$$

$$[\![\text{while } B \text{ do } C]\!] = \bigsqcup_{n \in \omega} F^n_{B,C}(\bot)$$

Observable Properties

properties $\mathcal{U} \subseteq D$

$\sqsubseteq$ information order $\Rightarrow$ if $x \in \mathcal{U}$, $x \sqsubseteq y$ then $y \in \mathcal{U}$

($\mathcal{U}$ upwards closed)

For $\Sigma = Var \to \mathbb{Z}$,

Observing = checking a variable

Observable = checkable on finitely many variables (semi-decidable)

e.g. $\{s : \Sigma \mid s(x) = 3\}$ ✓

$\{s : \Sigma \mid \forall x : Var.\ s(x) \text{ even}\}$ ✗

If $s = \bigsqcup_{i \in I} [x_i \mapsto n_i]$ (not directed, sneaky...)

and $s \in \mathcal{U} \Rightarrow \exists I' \subseteq_{fin} I.$

$\bigsqcup_{i \in I'} \langle x_i \mapsto n_i \rangle$

(inaccesible by directed joins) $\bigsqcup S \in \mathcal{U} \Rightarrow \exists s \in S.\ s \in \mathcal{U}$

Scott Topology

def $\mathcal{U} \subseteq D$ is Scott open if

- $\mathcal{U}$ upwards-closed
- $\bigsqcup S \in \mathcal{U} \Rightarrow \exists s \in S.\ s \in \mathcal{U}$

$\Omega D \triangleq \{\mathcal{U} \subseteq D \mid \mathcal{U} \text{ Scott open}\}$

$\bigcup_{i \in I} \mathcal{U}_i$ Scott open

$\bigcap_{i \in I_{fin}} \mathcal{U}_i$ Scott open

$\emptyset, D$ Scott open

why fin?

$$\bigsqcup S \in \bigcap_{i \in I_{fin}} \mathcal{U}_i \Rightarrow \forall i.\ \exists s_i \in S.\ s_i \in \mathcal{U}_i$$

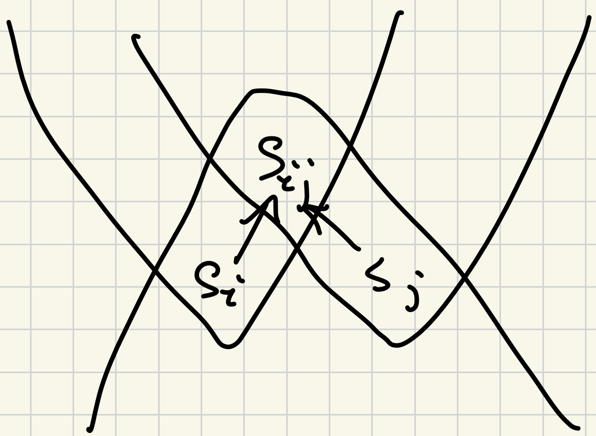
claim

$f : D \to E$ Scott-Continuous

iff

$f : (D, \Omega D) \to (E, \Omega E)$

Topologically continuous



Bridge elements

- $s \in \{s : \Sigma \mid s(x) = k\} \Longleftrightarrow s(x) = k \Longleftrightarrow [x \mapsto k] \sqsubseteq s$

  $\{s : \Sigma \mid s(x) = k\} = \uparrow [x \mapsto k]$ "prime" property

  $[x \mapsto k]$ "small" element

Scott-open:

$\bigsqcup S \in \uparrow [x \mapsto k] \Longleftrightarrow [x \mapsto k] \sqsubseteq \bigsqcup S \Longrightarrow \exists s \in S.\ [x \mapsto k] \sqsubseteq s$

$\Longleftrightarrow \exists s \in S.\ s \in \uparrow [x \mapsto k]$

- $\forall s : \Sigma.\quad s = \bigsqcup_{i \in I} \{[x \mapsto k] \sqsubseteq s\}$

  S described by small elems

  $\cong$

  $\mathcal{U}$ described by prime properties

  } Bridge between logic & programs

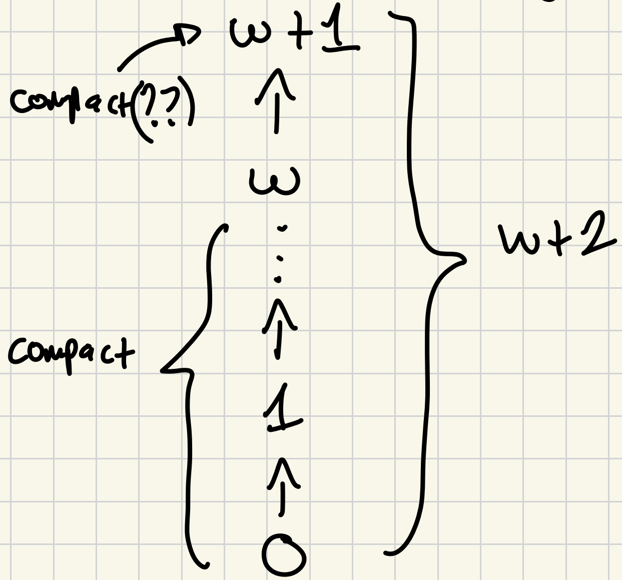
- Every $\mathcal{U} \in \Omega D$ is $\bigcup$nion of $\bigcap$ of $\uparrow [x \mapsto k]$

Compact Elements

def $c \in D$ is **compact** if $x \sqsubseteq \bigsqcup S \implies \exists y \in S.\ x \sqsubseteq y$ for all S directed

example

- $\bot$ is compact
- finite pfuncs $f : A \to B$,
- Every element in $A_\bot$
- $\omega + 1 \in \omega + 2$



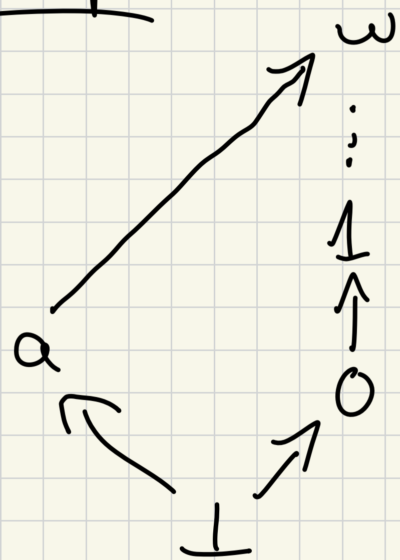
Notation

$KD = \{ c \in D \mid c \text{ compact} \}$

$\Downarrow d = \{ c \in KD \mid c \sqsubseteq d \}$

$c \ll d \overset{\Delta}{\iff} c \in \Downarrow d$

non-example



Is a compact? ✗

Is 1 compact? ✗

Algebraic DCPO

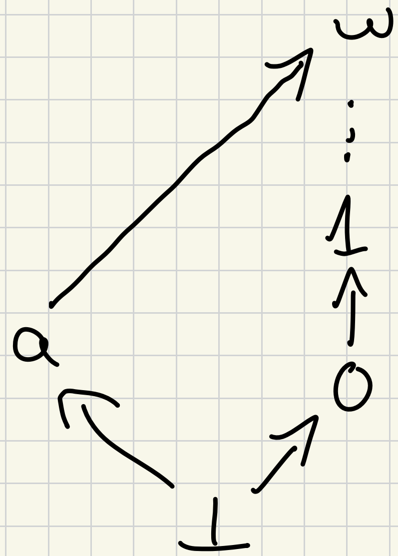
def D is *algebraic* if

$$\forall d \in D,\ \Downarrow d \text{ is directed, and } \bigsqcup \Downarrow d = d$$

example

- $A \to B$

non-example



$= D$

what is KD?

$\{\bot\}$

$\bigsqcup\{\bot\} = \bot \neq a \neq 0 \neq 1 \ldots$

theorem If D Algebraic, ΩD generated by $\{\uparrow c \mid c \in KD\}$, i.e.

proof $\mathcal{U} \in \Omega D \Rightarrow \mathcal{U} = \bigcup \{\uparrow c \mid c \in KD \;\&\; c \in \mathcal{U}\}$

$(\supseteq)$ upward-closure. $(\subseteq)$ $d \in \mathcal{U} \Leftrightarrow \bigsqcup \Downarrow d \in \mathcal{U} \Rightarrow \exists c \ll d.\ c \in \mathcal{U} \Rightarrow d \in \uparrow c$. $\square$

Algebraic Function Space

theorem? If D, E algebraic then $[D \to E]$ algebraic.

No, but for stupid reasons.

the band-aid fix

theorem If D alg and E Scott domain then $[D \to E]$ Scott.

$$K[D \to E] = \left\{ \bigsqcup_{i \in I_{fin}} \langle d_i \mapsto e_i \rangle \;\middle|\; \begin{array}{l} d_i, e_i \text{ compact} \\ \underbrace{\exists g. \forall i. \langle d_i \mapsto e_i \rangle \sqsubseteq g}_{\text{mutually consistent}} \end{array} \right.$$

$$\langle d_i \mapsto e_i \rangle (x) = \begin{cases} e_i & x \sqsupseteq d_i \\ \bot & \text{otherwise} \end{cases}$$

Logically, $f \in \uparrow \langle d_i \mapsto e_i \rangle$

$$f(\uparrow d_i) \subseteq \uparrow e_i \iff [\uparrow d_i]\ f\ [\uparrow e_i]$$

Hoare Triples!

Generic Rules

$$\frac{}{[\Phi]\ M\ [1]}\ \text{H-1} \qquad \frac{}{[0]\ M\ [\Psi]}\ \text{H-0}$$

$$\frac{[\Phi]\ M\ [\Psi] \qquad [\Phi]\ M\ [\Theta]}{[\Phi]\ M\ [\Psi \wedge \Theta]}\ \text{H-}\wedge \qquad \frac{[\Phi]\ M\ [\Psi] \qquad [\Theta]\ M\ [\Psi]}{[\Phi \vee \Theta]\ M\ [\Psi]}\ \text{H-}\vee \qquad \frac{\Phi \leq \Phi' \qquad [\Phi']\ M\ [\Psi'] \qquad \Psi' \leq \Psi}{[\Phi]\ M\ [\Psi]}\ \text{H-}\leq$$

Rules for Arithmetic Expressions

$$\frac{}{[\phi]\ k \in \mathbb{Z}\ [k]}\ \text{H-}\mathbb{Z} \qquad \frac{}{[\bigwedge_{i\in I}(n_i \rightsquigarrow k_i)]\ n_i\ [k_i]}\ \text{H-var-}\bigwedge$$

$$\frac{[\phi]\ A_1\ [\alpha] \qquad [\phi]\ A_2\ [\beta]}{[\phi]\ A_1\ \texttt{-}\ A_2\ [\alpha\ \texttt{-}\ \beta]}\ \text{H--} \qquad \frac{[\phi]\ A_1\ [\alpha] \qquad [\phi]\ A_2\ [\beta]}{[\phi]\ A_1\ \texttt{+}\ A_2\ [\alpha\ \texttt{+}\ \beta]}\ \text{H-+} \qquad \frac{[\phi]\ A_1\ [\alpha] \qquad [\phi]\ A_2\ [\beta]}{[\phi]\ A_1\ \texttt{*}\ A_2\ [\alpha\ \texttt{*}\ \beta]}\ \text{H-*}$$

This is a Hoare Logic because of previous theorem!

(It's not my fault)

Rules for Boolean Expressions

$$\frac{}{[\phi]\ \texttt{tt}\ [\texttt{tt}]}\ \text{H-}\texttt{tt} \qquad \frac{}{[\phi]\ \texttt{ff}\ [\texttt{ff}]}\ \text{H-}\texttt{ff}$$

$$\frac{[\phi]\ B\ [\chi]}{[\phi]\ \texttt{not}\ B\ [\texttt{not}\ \chi]}\ \text{H-}\texttt{not} \qquad \frac{[\phi]\ B_1\ [\chi] \quad [\phi]\ B_2\ [\kappa]}{[\phi]\ B_1\ \texttt{and}\ B_2\ [\chi\ \texttt{and}\ \kappa]}\ \text{H-}\texttt{and} \qquad \frac{[\phi]\ B_1\ [\chi] \quad [\phi]\ B_2\ [\kappa]}{[\phi]\ B_1\ \texttt{or}\ B_2\ [\chi\ \texttt{or}\ \kappa]}\ \text{H-}\texttt{or}$$

$$\frac{[\phi]\ A_1\ [\alpha] \quad [\phi]\ A_2\ [\beta]}{[\phi]\ A_1\ \texttt{=}\ A_2\ [\alpha\ \texttt{=}\ \beta]}\ \text{H-}\texttt{=} \qquad \frac{[\phi]\ A_1\ [\alpha] \quad [\phi]\ A_2\ [\beta]}{[\phi]\ A_1\ \texttt{<=}\ A_2\ [\alpha\ \texttt{<=}\ \beta]}\ \text{H-}\texttt{<=}$$

Rules for Commands

$$\frac{}{[\phi]\ \texttt{skip}\ [\phi]}\ \text{H-}\texttt{skip} \qquad \frac{[\phi]\ C_1\ [\theta] \quad [\theta]\ C_2\ [\psi]}{[\phi]\ C_1\texttt{;}C_2\ [\psi]}\ \text{H-}\texttt{;}$$

$$\frac{[\phi]\ B\ [\texttt{tt}] \quad [\phi]\ C_1\ [\psi]}{[\phi]\ \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\ [\psi]}\ \text{H-}\texttt{ite-tt} \qquad \frac{[\phi]\ B\ [\texttt{ff}] \quad [\phi]\ C_2\ [\psi]}{[\phi]\ \texttt{if}\ B\ \texttt{then}\ C_1\ \texttt{else}\ C_2\ [\psi]}\ \text{H-}\texttt{ite-ff}$$

$$\frac{[\phi]\ B\ [\texttt{tt}] \quad [\phi]\ C\ [\theta] \quad [\theta]\ \texttt{while}\ B\ \texttt{do}\ C\ [\psi]}{[\phi]\ \texttt{while}\ B\ \texttt{do}\ C\ [\psi]}\ \text{H-}\texttt{while-tt} \qquad \frac{[\phi]\ B\ [\texttt{ff}]}{[\phi]\ \texttt{while}\ B\ \texttt{do}\ C\ [\phi]}\ \text{H-}\texttt{while-ff}$$

$$\frac{[\phi]\ A\ [\alpha] \quad \mathsf{T}(\alpha)}{[\phi]\ \texttt{def}\ n\ \texttt{:=}\ A\ [\bigvee_{k\in\alpha}(n \rightsquigarrow k)]}\ \text{H-}\texttt{def} \qquad \frac{n_i \neq n}{[\bigwedge_{i\in I}(n_i \rightsquigarrow k_i)]\ \texttt{def}\ n\ \texttt{:=}\ A\ [n_i \rightsquigarrow k_i]}\ \text{H-}\texttt{def-}\wedge$$

Comparison with Sep. Logic (Semantics)

Definition 13 (Satisfiability Relation)**.** The *satisfiability relation* is a binary relation between logical states and assertions, written $e, s, h \vDash P$, given by

$$
\begin{array}{lcl}
e, s, h \vDash P_1 \wedge P_2 & \iff & e, s, h \vDash P_1 \wedge e, s, h \vDash P_2 \\
e, s, h \vDash P_1 \Rightarrow P_2 & \iff & e, s, h \vDash P_1 \implies e, s, h \vDash P_2 \\
e, s, h \vDash \mathsf{True} & \iff & \text{always} \\
e, s, h \vDash \mathsf{False} & \iff & \text{never} \\
e, s, h \vDash E_1 = E_2 & \iff & \mathcal{E}[\![E_1 = E_2]\!]_{e,s} = \mathsf{true} \\
e, s, h \vDash E_1 > E_2 & \iff & \mathcal{E}[\![E_1 > E_2]\!]_{e,s} = \mathsf{true} \\
e, s, h \vDash E \in X & \iff & \mathcal{E}[\![E \in X]\!]_{e,s} = \mathsf{true} \\
e, s, h \vDash \exists x.\, P & \iff & \exists v \in \mathsf{Val}.\, e[x \mapsto v], s, h \vDash P \\
e, s, h \vDash P_1 \star P_2 & \iff & \exists h_1, h_2 \in \mathsf{Heap}.\, h = h_1 \uplus h_2 \wedge e, s, h_1 \vDash P_1 \wedge e, s, h_2 \vDash P_2 \\
e, s, h \vDash P_1 \mathrel{-\!\!\star} P_2 & \iff & \forall h_1 \in \mathsf{Heap}.\, (h_1 \sharp h \wedge e, s, h_1 \vDash P_1) \implies e, s, h \uplus h_1 \vDash P_2 \\
e, s, h \vDash \mathsf{emp} & \iff & h = \emptyset \\
e, s, h \vDash \circledast_{E_1 \leq x < E_2} P & \iff & \mathcal{E}[\![E_1]\!]_{e,s} = i \in \mathbb{N} \wedge \mathcal{E}[\![E_2]\!]_{e,s} = k \in \mathbb{N} \wedge \\
 & & (i < k \implies \exists h_i, \ldots, h_{k-1}.\, h = h_i \uplus \ldots \uplus h_{k-1} \wedge \forall j.\, i \leq j < k.\, e, s, h_j \vDash P[j/x]) \wedge \\
 & & (i \geq k \implies h = \emptyset) \\
e, s, h \vDash E_1 \mapsto E_2 & \iff & h = \{\mathcal{E}[\![E_1]\!]_{e,s} \mapsto \mathcal{E}[\![E_2]\!]_{e,s}\}
\end{array}
$$

$\Rightarrow$ Not upwards closed on heaps h. i.e. $h \sqsubseteq h'$ & $e, s, h \vDash \mathsf{emp}$ $\not\Rightarrow$ $e, s, h' \vDash \mathsf{emp}$.

Sep. Logic more general than Scott semantics

Unexplored Avenues

- Reverse Process: Start with Logic $\Rightarrow$ Get deno. sem.?

$$\text{AlgDCPO} \xrightarrow{\simeq} \underbrace{\text{Super Coherent Space}}_{\text{conditions expressible on opens}}$$

"Stone Duality"

- Stone Duality for Separation Logic? (Simon Docherty, David Pym)
- Other Geometry-Algebra dualities in CS?
  - ↳ Algebraic Geometry for algebraic effects

References

- "Domain Theory in Logical Form" Abramsky 91 ☆
- "Domain Theory" Abramsky & Jung 95
- "Stone Spaces" Johnstone 82
- "Domains" Plotkin 83 (Also known as the "Pisa notes")
- "Mathematical Theory of Domains" Stoltenberg-Hansen, Lindström, Griffor 94
- "Topology via Logic" Vickers 89 ☆
- "Topological Characterization of Scott Domains" Sambin, Valentini (unpublished)