Denotational Semantics

Direct style denotational semantics

$\mathcal{S}_{ds} : \mathsf{Stm} \rightarrow (\mathsf{State} \hookrightarrow \mathsf{State})$ $\quad$ $\mathcal{S}_{ds}[S]$ is a partial function on states

Direct style denotational semantics

$\mathcal{S}_{ds} : \mathsf{Stm} \rightarrow (\mathsf{State} \hookrightarrow \mathsf{State})$ $\quad$ $\mathcal{S}_{ds}[S]$ is a partial function on states

$$\mathcal{S}_{ds}[x := a]s = s[x \mapsto \mathcal{A}[a]s]$$

Direct style denotational semantics

$\mathcal{S}_{ds} : \mathsf{Stm} \to (\mathsf{State} \hookrightarrow \mathsf{State})$ $\mathcal{S}_{ds}[S]$ is a partial function on states

$$\mathcal{S}_{ds}[x := a]s = s[x \mapsto \mathcal{A}[a]s]$$

$$\mathcal{S}_{ds}[\texttt{skip}] = \mathsf{id}$$

Direct style denotational semantics

$\mathcal{S}_{ds} : \mathsf{Stm} \to (\mathsf{State} \hookrightarrow \mathsf{State})$ $\mathcal{S}_{ds}[S]$ is a partial function on states

$$\mathcal{S}_{ds}[x := a]s = s[x \mapsto \mathcal{A}[a]s]$$

$$\mathcal{S}_{ds}[\mathtt{skip}] = \mathsf{id}$$

$$\mathcal{S}_{ds}[S_1; S_2] = \mathcal{S}_{ds}[S_2] \circ \mathcal{S}_{ds}[S_1]$$

Notation

$\mathsf{id}\ s = s$

$$(f \circ g)\ s = \begin{cases} f(g\ s) & \text{if } g\ s \neq \text{ undef} \\ & \text{and } f(g\ s) \neq \text{ undef} \\ \text{undef} & \text{otherwise} \end{cases}$$

Direct style denotational semantics

$\mathcal{S}_{ds} : \mathsf{Stm} \to (\mathsf{State} \hookrightarrow \mathsf{State})$ $\quad$ $\mathcal{S}_{ds}[S]$ is a partial function on states

$$\mathcal{S}_{ds}[x := a]s = s[x \mapsto \mathcal{A}[a]s]$$

$$\mathcal{S}_{ds}[\texttt{skip}] = \mathsf{id}$$

$$\mathcal{S}_{ds}[S_1; S_2] = \mathcal{S}_{ds}[S_2] \circ \mathcal{S}_{ds}[S_1]$$

$$\mathcal{S}_{ds}[\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2] = \\ \mathsf{cond}(\mathcal{B}[b], \mathcal{S}_{ds}[S_1], \mathcal{S}_{ds}[S_2])$$

Notation

$$\mathsf{cond}(p, g_1, g_2)\ s = \begin{cases} g_1\ s & \text{if } p\ s = \mathsf{tt} \\ & \text{and } g_1\ s \neq \mathsf{undef} \\ g_2\ s & \text{if } p\ s = \mathsf{ff} \\ & \text{and } g_2\ s \neq \mathsf{undef} \\ \mathsf{undef} & \text{otherwise} \end{cases}$$

$$p : \mathbf{State} \to \mathbb{T}$$

$$g_1, g_2 : \mathbf{State} \hookrightarrow \mathbf{State}$$

Direct style denotational semantics

$\mathcal{S}_{ds} : \mathsf{Stm} \to (\mathsf{State} \hookrightarrow \mathsf{State})$ $\mathcal{S}_{ds}[S]$ is a partial function on states

$$\mathcal{S}_{ds}[x := a]s = s[x \mapsto \mathcal{A}[a]s]$$

$$\mathcal{S}_{ds}[\texttt{skip}] = \mathsf{id}$$

$$\mathcal{S}_{ds}[S_1; S_2] = \mathcal{S}_{ds}[S_2] \circ \mathcal{S}_{ds}[S_1]$$

$$\mathcal{S}_{ds}[\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2] = \mathsf{cond}(\mathcal{B}[b],\, \mathcal{S}_{ds}[S_1],\, \mathcal{S}_{ds}[S_2])$$

$$\mathcal{S}_{ds}[\texttt{while } b \texttt{ do } S] = \text{?}$$

$$
\begin{aligned}
&\mathcal{S}_{ds}[\texttt{while}\ b\ \texttt{do}\ S] \\
&= \mathcal{S}_{ds}[\texttt{if}\ b\ \texttt{then}\ (S;\ \texttt{while}\ b\ \texttt{do}\ S) \\
&\qquad\quad \texttt{else skip}]
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{S}_{ds}[\texttt{while } b \texttt{ do } S] \\
&= \mathcal{S}_{ds}[\texttt{if } b \texttt{ then } (S; \texttt{ while } b \texttt{ do } S) \\
&\qquad\quad \texttt{else skip}] \\
&= \mathsf{cond}(\mathcal{B}[b], \mathcal{S}_{ds}[S; \texttt{ while } b \texttt{ do } S], \\
&\qquad\qquad \mathcal{S}_{ds}[\texttt{skip}]) \\
&= \mathsf{cond}(\mathcal{B}[b], \mathcal{S}_{ds}[\texttt{while } b \texttt{ do } S] \circ \mathcal{S}_{ds}[S], \\
&\qquad\qquad \mathsf{id}) \\
&= F(\mathcal{S}_{ds}[\texttt{while } b \texttt{ do } S])
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{S}_{ds}[\texttt{while } b \texttt{ do } S] \\
&= \mathcal{S}_{ds}[\texttt{if } b \texttt{ then } (S\texttt{; while } b \texttt{ do } S) \\
&\qquad \texttt{else skip}] \\
&= \mathsf{cond}(\mathcal{B}[b], \mathcal{S}_{ds}[S\texttt{; while } b \texttt{ do } S], \\
&\qquad\quad \mathcal{S}_{ds}[\texttt{skip}]) \\
&= \mathsf{cond}(\mathcal{B}[b], \mathcal{S}_{ds}[\texttt{while } b \texttt{ do } S] \circ \mathcal{S}_{ds}[S], \\
&\qquad\quad \mathsf{id}) \\
&= F(\mathcal{S}_{ds}[\texttt{while } b \texttt{ do } S])
\end{aligned}
$$

$$F : (\mathsf{State} \hookrightarrow \mathsf{State}) \rightarrow (\mathsf{State} \hookrightarrow \mathsf{State})$$

$$F = \lambda g.\mathrm{cond}(\mathcal{B}[b], g \circ \mathcal{S}_{ds}[S], id)$$

What is FIX?

FIX: ((State $\hookrightarrow$ State) $\rightarrow$ (State $\hookrightarrow$ State))

$\rightarrow$ (State $\hookrightarrow$ State)

$\mathcal{S}_{ds}[\![\texttt{while}\ b\ \texttt{do}\ S]\!]$

$= \mathcal{S}_{ds}[\![\texttt{if}\ b\ \texttt{then}\ (S;\ \texttt{while}\ b\ \texttt{do}\ S)$
$\quad\texttt{else skip}]\!]$

$= \text{cond}(\mathcal{B}[\![b]\!],\ \mathcal{S}_{ds}[\![S;\ \texttt{while}\ b\ \texttt{do}\ S]\!],$
$\quad\mathcal{S}_{ds}[\![\texttt{skip}]\!])$

$= \text{cond}(\mathcal{B}[\![b]\!],\ \mathcal{S}_{ds}[\![\texttt{while}\ b\ \texttt{do}\ S]\!] \circ \mathcal{S}_{ds}[\![S]\!],$
$\quad\text{id})$

$= F(\mathcal{S}_{ds}[\![\texttt{while}\ b\ \texttt{do}\ S]\!])$

$\mathcal{S}_{ds}[\![\texttt{while}\ b\ \texttt{do}\ S]\!]$ is a fixed point of F!

What is FIX?

$\mathcal{S}_{ds}[\texttt{while}\ b\ \texttt{do}\ S] = \mathsf{FIX}\ F$

where $F\ g = \mathsf{cond}(\mathcal{B}[b],\ g \circ \mathcal{S}_{ds}[S],\ \mathsf{id})$

Questions:

- will F always have a fixed point?
- could F have more than one fixed point? — which one do we choose?

Example

`while x`>0 `do skip`

$Fg = \mathrm{cond}(\mathsf{B}[\![x>0]\!], g \circ \mathrm{id}, \mathrm{id})$

Example

`while x`>0 `do skip`

$Fg = \text{cond}(\mathsf{B}[\![x>0]\!], g \circ \text{id}, \text{id})$

$g = \lambda s.\text{if } s\mathbf{x}>0 \text{ then } \mathbf{undef} \text{ else } s$

$h = \lambda s.s$

$Fg = g$ and $Fh = h \;\; \Rightarrow$
g and h are **both** possible solutions

Why g should be preferred to h?

Example

`while x`>0 `do skip`

$Fg = \text{cond}(\mathsf{B}[\![x>0]\!], g \circ \text{id}, \text{id})$

$g = \lambda s.\text{if } s\mathbf{x}>0 \text{ then } \mathbf{undef} \text{ else } s$

$h = \lambda s.s$

$Fg = g$ and $Fh = h \Rightarrow$
g and h are **both** possible solutions

Why g should be preferred to h?

g is less defined than h

Requirements to FIX

$\mathcal{S}_{ds}$[$\texttt{while}\ b\ \texttt{do}\ S$] = FIX F

where $F\ g$ = cond($\mathcal{B}[b]$, $g \circ \mathcal{S}_{ds}[S]$, id)

The desired fixed point FIX F should be some partial function g_0: State $\hookrightarrow$ State such that

Requirements to FIX

$\mathcal{S}_{ds}[\texttt{while}\ b\ \texttt{do}\ S] = \mathsf{FIX}\ F$

where $F\ g = \mathsf{cond}(\mathcal{B}[b],\ g \circ \mathcal{S}_{ds}[S],\ \mathsf{id})$

The desired fixed point FIX F should be some partial function g_0: State $\hookrightarrow$ State such that

- g_0 is a fixed point of F:

  $$F\ g_0 = g_0$$

- if g is another fixed point of F then g is at least as defined as g_0:

  if $F\ g = g$

  and $g_0\ s = s'$

  then $g\ s = s'$

  for all choices of s and s'.

An ordering on State $\hookrightarrow$ State

$$g_1 \sqsubseteq g_2$$

if and only if

if $g_1\ s = s'$ then $g_2\ s = s'$

for all choices of s and s'

An ordering on State $\hookrightarrow$ State

$$g_1 \sqsubseteq g_2$$

if and only if

$$\text{if } g_1\ s = s' \text{ then } g_2\ s = s'$$

for all choices of s and s'

Formalisation of requirements to FIX F

- FIX F is a fixed point of F

  FIX $F = F$ (FIX F)

- FIX F is the least fixed point of F

  if $g = F\ g$ then FIX $F \sqsubseteq g$

Example 5.6

Let g_1, g_2, g_3, and g_4 be partial functions in $\mathbf{State} \hookrightarrow \mathbf{State}$ defined as follows:

$$
\begin{aligned}
g_1\ s &= s \text{ for all } s \\
g_2\ s &= \begin{cases} s & \text{if } s\ \mathtt{x} \geq \mathbf{0} \\ \underline{\text{undef}} & \text{otherwise} \end{cases} \\
g_3\ s &= \begin{cases} s & \text{if } s\ \mathtt{x} = \mathbf{0} \\ \underline{\text{undef}} & \text{otherwise} \end{cases} \\
g_4\ s &= \begin{cases} s & \text{if } s\ \mathtt{x} \leq \mathbf{0} \\ \underline{\text{undef}} & \text{otherwise} \end{cases}
\end{aligned}
$$

Example 5.6

Let g_1, g_2, g_3, and g_4 be partial functions in $\mathbf{State} \hookrightarrow \mathbf{State}$ defined as follows:

$$
\begin{aligned}
g_1\ s &= s \text{ for all } s \\
g_2\ s &= \begin{cases} s & \text{if } s\ \texttt{x} \geq \mathbf{0} \\ \underline{\text{undef}} & \text{otherwise} \end{cases} \\
g_3\ s &= \begin{cases} s & \text{if } s\ \texttt{x} = \mathbf{0} \\ \underline{\text{undef}} & \text{otherwise} \end{cases} \\
g_4\ s &= \begin{cases} s & \text{if } s\ \texttt{x} \leq \mathbf{0} \\ \underline{\text{undef}} & \text{otherwise} \end{cases}
\end{aligned}
$$

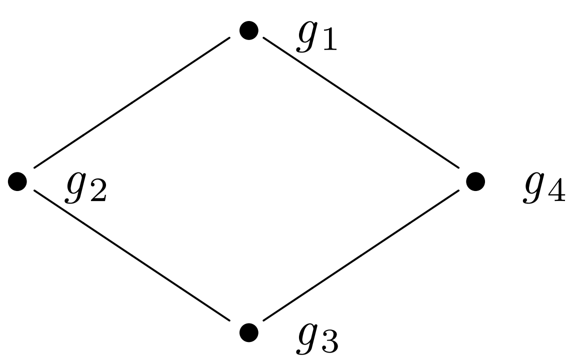
Then we have

$g_1 \sqsubseteq g_1$,

$g_2 \sqsubseteq g_1$, $g_2 \sqsubseteq g_2$,

$g_3 \sqsubseteq g_1$, $g_3 \sqsubseteq g_2$, $g_3 \sqsubseteq g_3$, $g_3 \sqsubseteq g_4$, and

$g_4 \sqsubseteq g_1$, $g_4 \sqsubseteq g_4$.



Partially ordered sets $(D, \sqsubseteq)$

A set D with an ordering $\sqsubseteq$ that is

- reflexive
  $d \sqsubseteq d$
- transitive
  $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_3$ imply $d_1 \sqsubseteq d_3$
- anti-symmetric
  $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_1$ imply $d_1 = d_2$

d is a least element of $(D, \sqsubseteq)$ if
$d \sqsubseteq d'$ for all d'

Fact 5.9

If a partially ordered set $(D, \sqsubseteq)$ has a least element d, then d is unique.

Proof: Assume that D has two least elements d_1 and d_2. Since d_1 is a least element, we have $d_1 \sqsubseteq d_2$. Since d_2 is a least element, we also have $d_2 \sqsubseteq d_1$.

The anti-symmetry of the ordering $\sqsubseteq$ then gives that $d_1 = d_2$. $\square$

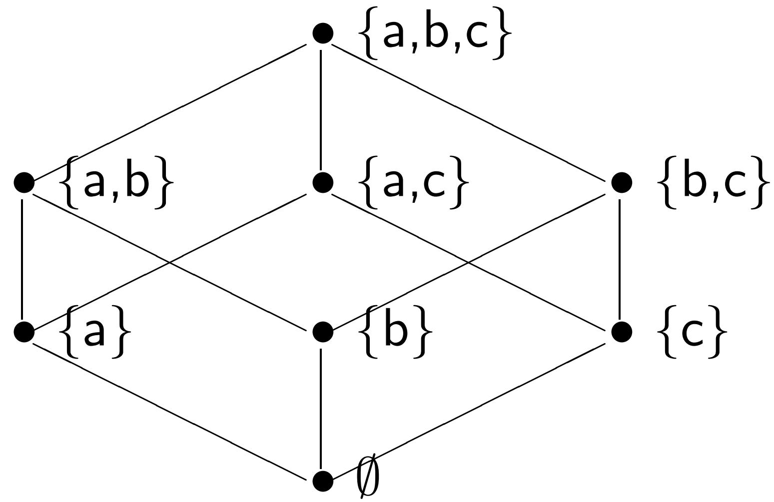
Example

Let $S \neq \emptyset$ and define

$$\mathcal{P}(S) = \{K \mid K \subseteq S\}$$

Then $(\mathcal{P}(S), \subseteq)$ is a partially ordered set.

If $S = \{$a,b,c$\}$ then the ordering is



The least element is $\emptyset$

$(\mathsf{State} \hookrightarrow \mathsf{State}, \sqsubseteq)$

Define the ordering $\sqsubseteq$ on State $\hookrightarrow$ State by

$$g_1 \sqsubseteq g_2$$

if and only if

$$\text{if } g_1\ s = s' \text{ then } g_2\ s = s'$$

for all choices of s and s'

Lemma 4.13

(State $\hookrightarrow$ State, $\sqsubseteq$) is a partially ordered set. The partial function $\perp$ defined by

$$\perp s = \text{undef for all } s$$

is the least element of State $\hookrightarrow$ State.

Example

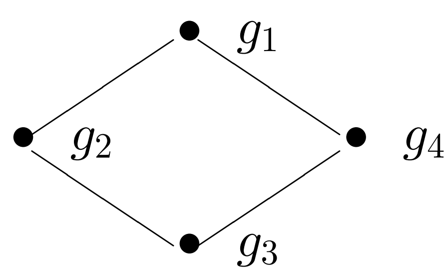
$g_1\ s = s$ for all s

$$g_2\ s = \begin{cases} s & \text{if } s\ x\ \geq\ 0 \\ \text{undef} & \text{otherwise} \end{cases}$$

$$g_3\ s = \begin{cases} s & \text{if } s\ x = 0 \\ \text{undef} & \text{otherwise} \end{cases}$$

$$g_4\ s = \begin{cases} s & \text{if } s\ x \leq 0 \\ \text{undef} & \text{otherwise} \end{cases}$$

The ordering



Upper bounds

Let $(D, \sqsubseteq)$ be a partially ordered set and let $Y \subseteq D$.

d is an upper bound on Y if

$d' \sqsubseteq d$ for all $d' \in Y$

d is a least upper bound on Y if

d is an upper bound on Y

if d' is an upper bound on Y
then $d \sqsubseteq d'$

Upper bounds

Let $(D, \sqsubseteq)$ be a partially ordered set and let $Y \subseteq D$.

d is an upper bound on Y if

$d' \sqsubseteq d$ for all $d' \in Y$

d is a least upper bound on Y if

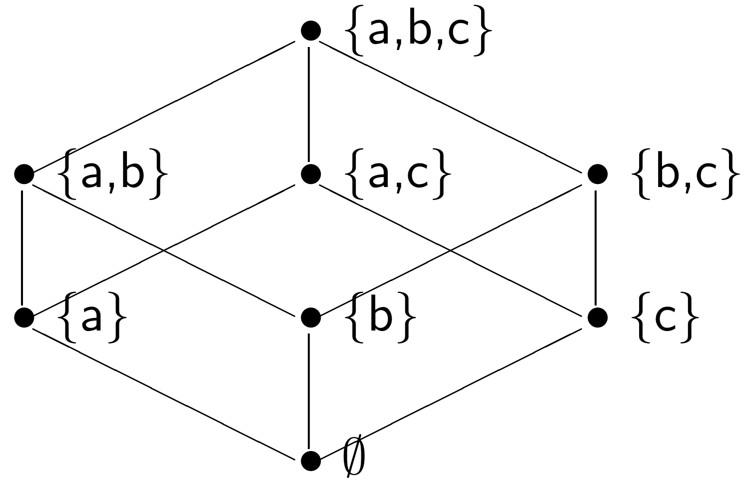
d is an upper bound on Y

if d' is an upper bound on Y
then $d \sqsubseteq d'$

Exercise 4.16

If Y has a least upper bound then it is unique and is denoted $\bigsqcup Y$

$(\mathcal{P}(\{a,b,c\}), \subseteq)$



$Y_0 = \{\ \emptyset, \{a\}, \{a,c\}\ \}$

$Y_1 = \{\ \emptyset, \{a\}, \{c\}, \{a,c\}\ \}$

$Y_2 = \{\ \}$

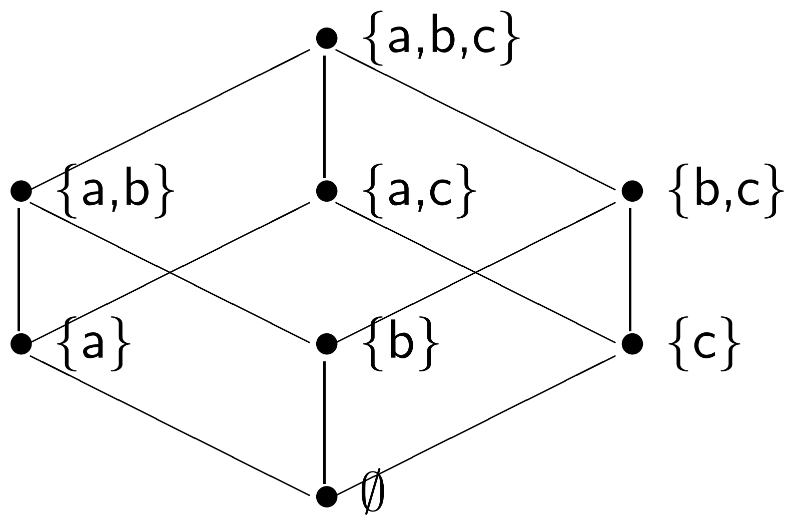
$Y_3 = \{\emptyset\}$ is a chain

$Y_4 = \{\ \{a\}, \{b, c\}\ \}$

Complete lattices

$(D, \sqsubseteq)$ is a complete lattice if every subset of D has a least upper bound

$(\mathcal{P}(\{a,b,c\}), \subseteq)$



is a complete lattice

Exercise 4.21

$(\mathsf{State} \hookrightarrow \mathsf{State}, \sqsubseteq)$ is not a complete lattice

A subset Y of D is called a <u>chain</u> if for any two elements d_1 and d_2 in Y either

$d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$

Chain complete partial ordered sets

$(D, \sqsubseteq)$ is a chain complete partially ordered set (ccpo) if every chain of D has a least upper bound

Lemma 4.25

(State $\hookrightarrow$ State, $\sqsubseteq$) is a chain complete partially ordered set.

The least upper bound $\bigsqcup Y$ of a chain Y is given by

$$(\bigsqcup Y)\ s = \begin{cases} g\ s & \text{if } g\ s \neq \text{ undef} \\ & \text{for some } g \in Y \\ \text{undef} & \text{otherwise} \end{cases}$$

Monotone functions

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be ccpo's and consider a (total) function

$$f : D \rightarrow D'$$

Then f is monotone if

whenever $d_1 \sqsubseteq d_2$ also $f\ d_1 \sqsubseteq' f\ d_2$

Examples

$$f_1, f_2 : \mathcal{P}(\{\mathsf{a,b,c}\}) \longrightarrow \mathcal{P}(\{\mathsf{d,e}\})$$

X	$f_1\ X$	$f_2\ X$
{a,b,c}	{d,e}	{d}
{a,b}	{d}	{d}
{a,c}	{d,e}	{d}
{b,c}	{d,e}	{e}
{a}	{d}	{d}
{b}	{d}	{e}
{c}	{e}	{e}
$\emptyset$	$\emptyset$	{e}

Monotone functions on CCPO's

Lemma 4.30

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be ccpo's and let

$$f : D \to D'$$

be a monotone function. If Y is a chain in D then $\{f\ d \mid d \in Y\}$ is a chain in D'. Furthermore,

$$\bigsqcup{}'\{f\ d \mid d \in Y\} \sqsubseteq' f(\bigsqcup Y)$$

Monotonicity is not enough

Example 4.31

$f : \mathcal{P}(\mathsf{N} \cup \{\mathsf{a}\}) \to \mathcal{P}(\mathsf{N} \cup \{\mathsf{a}\})$ defined by

$$f\ X = \begin{cases} X & \text{if } X \text{ is finite} \\ X \cup \{\mathsf{a}\} & \text{if } X \text{ is infinite} \end{cases}$$

Then f is a monotone function.

Monotonicity is not enough

Example 4.31

$f : \mathcal{P}(\mathsf{N} \cup \{\mathsf{a}\}) \to \mathcal{P}(\mathsf{N} \cup \{\mathsf{a}\})$ defined by

$$f\ X = \begin{cases} X & \text{if } X \text{ is finite} \\ X \cup \{\mathsf{a}\} & \text{if } X \text{ is infinite} \end{cases}$$

Then f is a monotone function.

But $\bigsqcup'\{f\ d \mid d \in Y\} = f(\bigsqcup Y)$ does not always hold.

Let $Y = \{\{0, 1, \cdots, n\} \mid n \geq 0\}$. Then

$$\bigsqcup\{f\ X \mid X \in Y\} = \bigsqcup Y = \mathsf{N}$$

But

$$f(\bigsqcup Y) = f\ \mathsf{N} = \mathsf{N} \cup \{\mathsf{a}\}$$

Continuous functions

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be ccpo's and consider a (total) function $f : D \to D'$. Then f is continuous if

- f is monotone
- $\bigsqcup'\{f\ d \mid d \in Y\} = f\ (\bigsqcup Y)$

for all non-empty chains Y of D

Exercise 4.34

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be ccpo's and let

$$f : D \to D'$$

be a (total) function satisfying

$$\bigsqcup'\{f\ d \mid d\ \in\ Y\} = f\ (\bigsqcup Y)$$

for all non-empty chains Y of D.
Then f is monotone

Example: Factorial program

$$\mathcal{S}_{ds}[\texttt{y := 1; while } \neg(\texttt{x}=1) \texttt{ do } (\texttt{y:=y}\star\texttt{x; x:=x}-1)]\ s$$

$$= (\mathsf{FIX}\ F)\ (s[\texttt{y} \mapsto 1])$$

where

$$(F\ g)\ s = \begin{cases} g(s[\texttt{y} \mapsto (s\ \texttt{y} \star s\ \texttt{x})]\ \ [\texttt{x} \mapsto (s\ \texttt{x}) - 1]) & \text{if } s\ \texttt{x} \neq 1 \\ s & \text{if } s\ \texttt{x} = 1 \end{cases}$$

Then F is continuous

Fixed point theorem

Theorem 4.37

Let $f : D \rightarrow D$ be a continuous function on the ccpo $(D, \sqsubseteq)$ with least element $\perp$. Then

$$\text{FIX } f = \bigsqcup\{f^n \perp \mid n \geq 0\}$$

defines an element of D and this element is the least fixed point of f.

Notation:

$$f^0 = \text{id}$$
$$f^{n+1} = f \circ f^n \text{ for } n \geq 0$$

Proof: We first show the *well-definedness* of FIX f. Note that $f^0 \perp = \perp$ and that $\perp \sqsubseteq d$ for all $d \in D$. By induction on n, one may show that

$$f^{\mathrm{n}} \perp \sqsubseteq f^{\mathrm{n}}\ d$$

for all $d \in D$ since f is monotone. It follows that $f^{\mathrm{n}} \perp \sqsubseteq f^{\mathrm{m}} \perp$ whenever $\mathrm{n} \leq \mathrm{m}$. Hence $\{ f^{\mathrm{n}} \perp \mid \mathrm{n} \geq 0 \}$ is a (non-empty) chain in D, and FIX f exists because D is a ccpo.

We next show that FIX f is a *fixed point*; that is, f (FIX f) = FIX f. We calculate

$$\begin{array}{rcll}
f\ (\mathsf{FIX}\ f) & = & f\ (\bigsqcup \{ f^{\mathrm{n}} \perp \mid \mathrm{n} \geq 0 \}) & (\text{definition of } \mathsf{FIX}\ f) \\
 & = & \bigsqcup \{ f(f^{\mathrm{n}} \perp) \mid \mathrm{n} \geq 0 \} & (\text{continuity of } f) \\
 & = & \bigsqcup \{ f^{\mathrm{n}} \perp \mid \mathrm{n} \geq 1 \} & \\
 & = & \bigsqcup (\{ f^{\mathrm{n}} \perp \mid \mathrm{n} \geq 1 \} \cup \{\perp\}) & (\bigsqcup (Y \cup \{\perp\}) = \bigsqcup Y \\
 & & & \text{for all chains } Y) \\
 & = & \bigsqcup \{ f^{\mathrm{n}} \perp \mid \mathrm{n} \geq 0 \} & (f^0 \perp = \perp) \\
 & = & \mathsf{FIX}\ f & (\text{definition of } \mathsf{FIX}\ f)
\end{array}$$

To see that FIX f is the *least* fixed point, assume that d is some other fixed point. Clearly $\perp \sqsubseteq d$ so the monotonicity of f gives $f^{\mathrm{n}} \perp \sqsubseteq f^{\mathrm{n}}\ d$ for $\mathrm{n} \geq 0$, and as d was a fixed point, we obtain $f^{\mathrm{n}} \perp \sqsubseteq d$ for all $\mathrm{n} \geq 0$. Hence d is an upper bound of the chain $\{ f^{\mathrm{n}} \perp \mid \mathrm{n} \geq 0 \}$, and using that FIX f is the least upper bound, we have FIX $f \sqsubseteq d$. □

Example: Factorial program

$$\mathcal{S}_{ds}[\texttt{y := 1; while } \neg(\texttt{x}=\texttt{1}) \texttt{ do} \\ (\texttt{y:=y}\star\texttt{x; x:=x}-\texttt{1})]\ s$$

$$= (\text{FIX } F)\ (s[\texttt{y} \mapsto 1])$$

where

$$(F\ \ g)\ s \\ = \begin{cases} g(s[\texttt{y} \mapsto (s\ \ \texttt{y} \star s\ \ \texttt{x})]\ \ [\texttt{x} \mapsto (s\ \ \texttt{x}) - 1]) & \\ & \text{if } s\ \texttt{x}\ \neq\ 1 \\ s & \text{if } s\ \texttt{x} = 1 \end{cases}$$

We must ensure that

- (State $\hookrightarrow$ State, $\sqsubseteq$) is a ccpo
  Lemma 4.25
- F is a continuous function

Then Theorem 4.37 can be applied

$$F\ g\ s = \begin{cases} g\ (s[\mathbf{y}\mapsto(s\ \mathbf{y})\cdot(s\ \mathbf{x})][\mathbf{x}\mapsto(s\ \mathbf{x})-\mathbf{1}]) & \text{if } s\ \mathbf{x} \neq \mathbf{1} \\ s & \text{if } s\ \mathbf{x} = \mathbf{1} \end{cases}$$

Example: Factorial program

$$(F^0 \bot)s = \text{undef}$$

$$(F^1 \bot)s = \begin{cases} \text{undef} & \text{if } s\ \mathtt{x} \neq 1 \\ s & \text{if } s\mathtt{x} = 1 \end{cases}$$

$$(F^2 \bot)s$$

$$= \begin{cases} \text{undef} & \text{if } s\ \mathtt{x} \neq 1 \text{and } s\ \mathtt{x} \neq 2 \\ s[\mathtt{y} \mapsto (s\ \mathtt{y}) \star 2][\mathtt{x} \mapsto 1] & \\ & \text{if } s\ \mathtt{x} = 2 \\ s & \text{if } s\ \mathtt{x} = 1 \end{cases}$$

$$(F^n \bot)s$$

$$= \begin{cases} \text{undef} & \text{if } s\ \mathtt{x} < 1 \text{ or } s\ \mathtt{x} > n \\ s[\mathtt{y} \mapsto (s\ \mathtt{y}) \star j \star \ldots \star 2 \star 1][\mathtt{x} \mapsto 1] & \\ & \text{if } s\ \mathtt{x} = j \text{ and } 1 \leq j \leq n \end{cases}$$

$$(\text{FIX } F)s$$

$$= \begin{cases} \text{undef} & \text{if } s\ \mathtt{x} < 1 \\ s[\mathtt{y} \mapsto (s\ \mathtt{y}) \star n \star \ldots \star 2 \star 1][\mathtt{x} \mapsto 1] & \\ & \text{if } s\ \mathtt{x} = n \text{ and } 1 \leq n \end{cases}$$

Direct style denotational semantics

$\mathcal{S}_{ds}$: Stm $\rightarrow$ (State $\hookrightarrow$ State)

$\mathcal{S}_{ds}[x := a]s = s[x \mapsto \mathcal{A}[a]s]$

$\mathcal{S}_{ds}[\texttt{skip}] = \mathsf{id}$

$\mathcal{S}_{ds}[S_1; S_2] = \mathcal{S}_{ds}[S_2] \circ \mathcal{S}_{ds}[S_1]$

$\mathcal{S}_{ds}[\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2] =$
$\qquad \mathsf{cond}(\mathcal{B}[b], \mathcal{S}_{ds}[S_1], \mathcal{S}_{ds}[S_2])$

$\mathcal{S}_{ds}[\texttt{while } b \texttt{ do } S] = \mathsf{FIX}\ F$
where
$\qquad F\ g = \mathsf{cond}(\mathcal{B}[b], g \circ \mathcal{S}_{ds}[S], \mathsf{id})$

Well-definedness of $\mathcal{S}_{ds}$

Proposition 4.47

The semantic equations for $\mathcal{S}_{ds}$ define a total function in $\mathsf{Stm} \rightarrow (\mathsf{State} \hookrightarrow \mathsf{State})$.

Auxiliary results

Lemma 4.43

Let g_0: State $\hookrightarrow$ State, p: State $\to$ T and define

$$F\ g = \mathrm{cond}(p, g, g_0)$$

Then F is continuous.

Lemma 4.45

Let g_0: State $\hookrightarrow$ State and define

$$F\ g = g \circ g_0$$

Then F is continuous.

Lemma 4.35

If $f : D \to D'$ and $f' : D' \to D''$ are continuous functions then $f' \circ f$ is a continuous function

Proofs of Lemma 4.43 and 4.45 are easy.

Lemma 4.43: first show that F is monotone, thus obtaining one inequality, and then prove the remaining inequality

Lemma 4.45: just write it.

Proofs of Lemma 4.43 and 4.45 are easy.

Lemma 4.43: first show that F is monotone, thus obtaining one inequality, and then prove the remaining inequality

Lemma 4.45: just write it.

Hence

$$F\ g = \mathsf{cond}(\mathcal{B}[b],\ g \circ \mathcal{S}_{ds}[S],\ \mathsf{id})$$

is continuous

Fixpoint induction lemma

Exercise 5.40 (Essential)

Let $f\colon D \to D$ be a continuous function on a ccpo $(D, \sqsubseteq)$ and let $d \in D$ satisfy $f\ d \sqsubseteq d$. Show that $\mathsf{FIX}\ f \sqsubseteq d$. $\square$

One such d is called a pre-fixpoint.

Denotational semantics of repeat-until

Denotational semantics of repeat-until

$\texttt{repeat}\ S\ \texttt{until}\ b \cong$ S; $\texttt{if}\ b\ \texttt{then skip else repeat}\ S\ \texttt{until}\ b$

Thus we have:

$\mathcal{S}[\![\texttt{repeat}\ S\ \texttt{until}\ b]\!] = \text{cond}(\mathcal{B}[\![b]\!], \text{id}, \mathcal{S}[\![\texttt{repeat}\ S\ \texttt{until}\ b]\!]) \circ \mathcal{S}[\![S]\!]$

Functional $Fg = \text{cond}(\mathcal{B}[\![b]\!], \text{id}, g) \circ \mathcal{S}[\![S]\!]$ is continuous so that

$\mathcal{S}[\![\texttt{repeat}\ S\ \texttt{until}\ b]\!] = \text{FIX}\ F$

Denotational semantics of for

Denotational semantics of for

$\texttt{for x:=}a_1 \texttt{ to } a_2 \texttt{ do } S \;\cong\; \texttt{x:=}a_1\texttt{; while x}\leq a_2 \texttt{ do } (S\texttt{; x:=x+1})$

Thus its denotational semantics relies on the denotational semantics of

$\texttt{while x}\leq a_2 \texttt{ do } (S\texttt{; x:=x+1})$

Equivalence in denotational semantics

$$P \cong Q \text{ when } \mathcal{S}[\![P]\!] = \mathcal{S}[\![Q]\!]$$

Equivalence in denotational semantics

$$\mathrm{P} \cong \mathrm{Q} \text{ when } \mathcal{S}[\![\mathrm{P}]\!] = \mathcal{S}[\![\mathrm{Q}]\!]$$

Easy examples:

S; skip $\cong$ S

$S_1;(S_2;S_3) \cong (S_1;S_2);S_3$

`while` b `do` S $\cong$ `if` b `then` (S`;while` b `do` S) `else skip`

Not easy: $\texttt{repeat}\ S\ \texttt{until}\ b \ \cong_{\text{ds}}\ S; \texttt{while}\ \neg b\ \texttt{do}\ S$

Not easy: $\texttt{repeat}\ S\ \texttt{until}\ b \cong_{\text{ds}} S; \texttt{while}\ \neg b\ \texttt{do}\ S$

$$\mathcal{S}_{ds}[\![\texttt{repeat}\ S\ \texttt{until}\ b]\!] = \text{FIX}\ F_1 \text{ where}$$

$$F_1\, g = cond(\mathcal{B}[\![b]\!], id, g) \circ \mathcal{S}_{ds}[\![S]\!]$$

$$\mathcal{S}_{ds}[\![S; \texttt{while}\ \neg b\ \texttt{do}\ S]\!] = \text{FIX}\ F_2 \circ \mathcal{S}_{ds}[\![S]\!] \text{ where}$$

$$\begin{aligned} F_2\, g &= cond(\mathcal{B}[\![\neg b]\!], g \circ \mathcal{S}_{ds}[\![S]\!], id) \\ &= cond(\mathcal{B}[\![b]\!], id, g \circ \mathcal{S}_{ds}[\![S]\!]) \end{aligned}$$

Consider generic state function g_0 and state predicate p so that

$$F_1 = \lambda g.cond(p, id, g) \circ g_0 \quad \text{and} \quad F_2 = \lambda g.cond(p, id, g \circ g_0)$$

(A) FIX $F_1 \sqsubseteq (\text{FIX } F_2) \circ g_0$. It is easily seen that $(\text{FIX } F_2) \circ g_0$ is a pre-fixpoint of F_1.

(A) $\mathrm{FIX}\ F_1 \sqsubseteq (\mathrm{FIX}\ F_2) \circ g_0$. **It is easily seen that** $(\mathrm{FIX}\ F_2) \circ g_0$ **is a pre-fixpoint of** F_1**.**

$$F_1((\mathrm{FIX}\,F_2) \circ g_0) \sqsubseteq (\mathrm{FIX}\,F_2) \circ g_0 \ \Rightarrow\ \mathrm{FIX}\,F_1 \sqsubseteq (\mathrm{FIX}\,F_2) \circ g_0$$

$$cond(p, id, (\mathrm{FIX}\,F_2) \circ g_0) \circ g_0 \sqsubseteq (\mathrm{FIX}\,F_2) \circ g_0$$

This holds because
$(\mathrm{FIX}\,F_2) \circ g_0 = F_2((\mathrm{FIX}\,F_2)) \circ g_0 = cond(p, id, (\mathrm{FIX}\,F_2) \circ g_0) \circ g_0$

(B) $(\text{FIX } F_2) \circ g_0 \sqsubseteq \text{FIX } F_1$. By fixpoint theorem, $\text{FIX } F_2 = \bigvee_{n \geq 0} F_2^n \bot$. Since $\lambda g.g \circ g_0$ is continuous (exercise), we have that

$$(\text{FIX } F_2) \circ g_0 = (\bigvee_{n \geq 0} F_2^n \bot) \circ g_0 = \bigvee_{n \geq 0} ((F_2^n \bot) \circ g_0).$$

(B) $(\text{FIX } F_2) \circ g_0 \sqsubseteq \text{FIX } F_1$. By fixpoint theorem, $\text{FIX } F_2 = \bigvee_{n \geq 0} F_2^n \bot$. Since $\lambda g. g \circ g_0$ is continuous (exercise), we have that

$$(\text{FIX } F_2) \circ g_0 = (\bigvee_{n \geq 0} F_2^n \bot) \circ g_0 = \bigvee_{n \geq 0} ((F_2^n \bot) \circ g_0).$$

Let us show by induction on $n \geq 0$ that $\forall n \geq 0. (F_2^n \bot) \circ g_0 \sqsubseteq \text{FIX } F_1$.

$(n = 0)$ $(F_2^0 \bot) \circ g_0 = \bot \circ g_0 = \bot \sqsubseteq \text{FIX } F_1$.

(B) (FIX F_2) $\circ\, g_0 \sqsubseteq$ FIX F_1. By fixpoint theorem, FIX $F_2 = \vee_{n\geq 0} F_2^n \bot$. Since $\lambda g.g \circ g_0$ is continuous (exercise), we have that

$$(\text{FIX } F_2) \circ g_0 = (\vee_{n\geq 0} F_2^n \bot) \circ g_0 = \vee_{n\geq 0}((F_2^n \bot) \circ g_0).$$

Let us show by induction on $n \geq 0$ that $\forall n \geq 0.(F_2^n \bot) \circ g_0 \sqsubseteq$ FIX F_1.

$(n = 0)$ $(F_2^0 \bot) \circ g_0 = \bot \circ g_0 = \bot \sqsubseteq$ FIX F_1.

$(n+1)$ $(F_2^{n+1} \bot) \circ g_0 = (F_2(F_2^n \bot)) \circ g_0 = cond(p, id, (F_2^n \bot) \circ g_0) \circ g_0$.

Also: FIX $F_1 = F_1(\text{FIX } F_1) = cond(p, id, \text{FIX } F_1) \circ g_0$.

By inductive hypothesis, $(F_2^n \bot) \circ g_0 \sqsubseteq$ FIX F_1.

Thus, by continuity of $\lambda g.g \circ g_0$ and $\lambda g.cond(p, h, g)$, we have that

$$cond(p, id, (F_2^n \bot) \circ g_0) \circ g_0 \sqsubseteq cond(p, id, \text{FIX } F_1) \circ g_0$$

Equivalence

Theorem 4.55

For every statement S of While we have

$$\mathcal{S}_{sos}[S] = \mathcal{S}_{ds}[S]$$

where

$$\mathcal{S}_{sos}[S]\ s = \begin{cases} s' & \text{if } (S, s) \Rightarrow^* s' \\ \text{undefined} & \text{otherwise} \end{cases}$$

Proof:

Lemma 4.56:

For every statement S of While we have

$$\mathcal{S}_{sos}[S] \sqsubseteq \mathcal{S}_{ds}[S]$$

⇐ We show this as exercise

Lemma 4.57:

For every statement S of While we have

$$\mathcal{S}_{ds}[S] \sqsubseteq \mathcal{S}_{sos}[S]$$

Auxiliary results: conditional

Lemma 4.43

Let g_0: State $\hookrightarrow$ State, p: State $\rightarrow$ T and define

$$F\ g = \mathrm{cond}(p, g, g_0)$$

Then F is continuous.

Exercise 4.44

Let g_0: State $\hookrightarrow$ State, p: State $\rightarrow$ T and define

$$F\ g = \mathrm{cond}(p, g_0, g)$$

Then F is continuous.

Auxiliary results: composition

Lemma 4.45

Let g_0: State $\hookrightarrow$ State and define

$$F\ g = g \circ g_0$$

Then F is continuous.

Exercise 4.46

Let g_0: State $\hookrightarrow$ State and define

$$F\ g = g_0 \circ g$$

Then F is continuous.

Lemma 4.56. $\forall S . \mathcal{S}_{sos}[\![S]\!] \sqsubseteq \mathcal{S}_{ds}[\![S]\!]$.

Lemma 4.56. $\forall S. \mathcal{S}_{sos}[\![S]\!] \sqsubseteq \mathcal{S}_{ds}[\![S]\!]$.

Proof.

In order to show that $\langle S, s\rangle \Rightarrow^* s'$ implies $\mathcal{S}_{ds}[\![S]\!]s = s'$, we prove:

(A) $\langle S, s\rangle \Rightarrow s'$ implies $\mathcal{S}_{ds}[\![S]\!]s = s'$
(B) $\langle S, s\rangle \Rightarrow \langle S', s'\rangle$ implies $\mathcal{S}_{ds}[\![S]\!]s = \mathcal{S}_{ds}[\![S']\!]s'$

From (A) and (B): $\langle S, s\rangle \Rightarrow^* s'$ iff $\exists k. \langle S, s\rangle \Rightarrow^k \langle S_0, s_0\rangle \Rightarrow s'$. By (A), $\mathcal{S}_{ds}[\![S_0]\!]s_0 = s'$. By applying (B) $k \geq 0$ times, $\mathcal{S}_{ds}[\![S]\!]s = \mathcal{S}_{ds}[\![S_0]\!]s_0$. Hence $\mathcal{S}_{ds}[\![S]\!]s = s'$.

(A) is shown by structural induction on S: only straightforward bases cases for assignment and skip, because the inductive steps vacuously hold.

(B) is shown by structural induction on S. The base cases for assignment and skip vacuously hold.

Inductive steps.
$(S_1; S_2)$ Two possibilities for $\langle S_1; S_2, s\rangle \Rightarrow \langle S', s'\rangle$

(1) $\langle S_1; S_2, s\rangle \Rightarrow \langle S_1'; S_2, s'\rangle$ where $\langle S_1, s\rangle \Rightarrow \langle S_1', s'\rangle$. By induction on S_1, $\mathcal{S}_{ds}[\![S_1]\!]s = \mathcal{S}_{ds}[\![S_1']\!]s'$, so that

$$\mathcal{S}_{ds}[\![S_1; S_2]\!]s = \mathcal{S}_{ds}[\![S_2]\!](\mathcal{S}_{ds}[\![S_1]\!]s) =$$

$$= \mathcal{S}_{ds}[\![S_2]\!](\mathcal{S}_{ds}[\![S_1']\!]s') = \mathcal{S}_{ds}[\![S_1'; S_2]\!]s'$$

(2) $\langle S_1; S_2, s\rangle \Rightarrow \langle S_2, s'\rangle$ where $\langle S_1, s\rangle \Rightarrow s'$. Then, by (A), $\mathcal{S}_{ds}[\![S_1]\!]s = s'$, so that $\mathcal{S}_{ds}[\![S_1; S_2]\!]s = \mathcal{S}_{ds}[\![S_2]\!]s'$.

($\texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2$): easy.

($\texttt{while}\ b\ \texttt{do}\ S$): the only possibility is

$$\langle \texttt{while}\ b\ \texttt{do}\ S, s\rangle \Rightarrow$$
$$\langle \texttt{if}\ b\ \texttt{then}\ (S; \texttt{while}\ b\ \texttt{do}\ S)\ \texttt{else}\ \texttt{skip}, s\rangle$$

Thus, this is a consequence of $\mathcal{S}_{ds}[\![\texttt{while}\ b\ \texttt{do}\ S]\!] = \mathcal{S}_{ds}[\![\texttt{if}\ b\ \texttt{then}\ (S; \texttt{while}\ b\ \texttt{do}\ S)\ \texttt{else}\ \texttt{skip}]\!]$.