$[\![\textbf{while } B \textbf{ do } C]\!] \in (State \to State)$

fixed point.

say of a function(al)

$f_{[\![B]\!],[\![C]\!]}$

$: (State \to State) \to (State \to State)$

NB Fixed points are fixed values of endo functions; that is, function from a set to itself.

$[\![\textbf{while } B \textbf{ do } C]\!]$

$$[\![\text{while } B \text{ do } C]\!] = \lambda s.\ \text{if}\left([\![B]\!]s,\ [\![\text{while } B \text{ do } C]\!]([\![C]\!]s),\ s\right)$$

$$\stackrel{?}{=} f_{[\![B]\!],[\![C]\!]}\left([\![\text{while } B \text{ do } C]\!]\right)$$

def

$$\lambda w \in (State \to State).\ \lambda s.\ \text{if}\left([\![B]\!]s,\ w([\![C]\!]s),\ s\right)$$

We want

$$[\![\text{while } B \text{ do } C]\!] = \underline{fix}\left(f_{[\![B]\!],[\![C]\!]}\right)$$

$[\![\textbf{while } B \textbf{ do } C]\!]$

Justify for!

Example: $B = \text{true}$, $C = \text{skip}$

Intuitively $[\![\text{while true do skip}]\!]$ $\in (\text{State} \rightharpoonup \text{State})$

$=$ the totally undefined partial function

$\stackrel{\text{def}}{=} \perp \sim$ the function whose graph is the empty set

$$f_{[\![\text{true}]\!],[\![\text{skip}]\!]}$$

$$= \lambda w \in (\text{State} \rightharpoonup \text{State}).\ \lambda s \in \text{State}.\ \text{if}(\text{true}, w(s), s)$$

$$= \lambda w.\, \lambda s.\, w(s) = \lambda w.\, w = \text{identity}$$

$[\![\textbf{while } B \textbf{ do } C]\!]$

$$[\![\text{while true do skip}]\!] = \underline{fix}\,(\lambda w \in (State \to State).\, w)$$

INTUITIVELY $\longrightarrow$ $= \perp \in (States \to States)$

$\boxed{?}$ What should $\underline{fix}$ be? So that it matches the operational behaviour.

$\boxed{?}$ How can we characterise it mathematically?

<u>NB</u>: A, $\lambda x\, x : A \to A$

Every element of A is a fixed point.

$[\![\textbf{while } B \textbf{ do } C]\!] \in (State \to State)$

$$[\![\text{while } B \text{ do } C]\!]_0 \stackrel{\text{def}}{=} \perp$$

$$[\![\text{while } B \text{ do } C]\!]_1 \stackrel{\text{def}}{=} f_{[\![B]\!],[\![C]\!]}([\![\text{while } B \text{ do } C]\!]_0)$$

$$= f_{[\![B]\!],[\![C]\!]}(\perp)$$

notation for undefined

$$= \lambda s.\ if([\![B]\!]s, \uparrow, s)$$

$$= \lambda s. \begin{cases} s & [\![B]\!]s = false \\ \uparrow & otw \end{cases}$$

$[\![\textbf{while } B \textbf{ do } C]\!]$

$$[\![\text{while } B \text{ do } C]\!]_2 = f_{[\![B]\!],[\![C]\!]}\left([\![\text{while } B \text{ do } C]\!]_1\right)$$

$$= \lambda s.\ \text{if}\left([\![B]\!]s,\ [\![\text{while } B \text{ do } C]\!]_1([\![C]\!]s),\ s\right)$$

$$= \lambda s. \begin{cases} s & \text{if } [\![B]\!]s = \text{false} \\ [\![\text{while } B \text{ do } C]\!]_1([\![C]\!]s) & \text{otw} \end{cases}$$

$$= \lambda s. \begin{cases} s & \text{if } [\![B]\!]s = \text{false} \\ [\![C]\!](s) & \text{if } [\![B]\!]s = \text{true} \ \&\ [\![B]\!]([\![C]\!]s) = \text{false} \\ \uparrow & \text{otw} \end{cases}$$

$[\![\textbf{while } B \textbf{ do } C]\!]$

In general

$$[\![\text{while } B \text{ do } C]\!]_{n+1} = f_{[\![B]\!],[\![C]\!]}\left([\![\text{while } B \text{ do } C]\!]_n\right)$$

gives information about going through the loop n times.

$[\![\text{while } B \text{ do } C]\!]_0 = \bot$ NO INFORMATION

$\sqsubseteq f_{[\![B]\!],[\![C]\!]}\left([\![\text{while } B \text{ do } C]\!]_0\right) = [\![\text{while } B \text{ do } C]\!]_1$

NOTION OF INFORMATION $\sqsubseteq [\![\text{while } B \text{ do } C]\!]_n \sqsubseteq [\![\text{while } B \text{ do } C]\!]_{n+1} \sqsubseteq$

$[\![\textbf{while } B \textbf{ do } C]\!]$

Finally

$$[\![\text{while } B \text{ do } C]\!] = \bigsqcup_{n \in \mathbb{N}} [\![\text{while } B \text{ do } C]\!]_n$$

– PASSING TO THE LIMIT

– PUTTING ALL THE INFORMATION TOGETHER.

Fixed point property of $[\![\textbf{while } B \textbf{ do } C]\!]$

$$[\![\textbf{while } B \textbf{ do } C]\!] = f_{[\![B]\!],[\![C]\!]}([\![\textbf{while } B \textbf{ do } C]\!])$$

where, for each $b : State \to \{true, false\}$ and $c : State \rightharpoonup State$, we define

$$f_{b,c} : (State \rightharpoonup State) \to (State \rightharpoonup State)$$

as

$$f_{b,c} = \lambda w \in (State \rightharpoonup State).\, \lambda s \in State.\, if\big(b(s), w(c(s)), s\big).$$

Fixed point property of $[\![\mathbf{while}\ B\ \mathbf{do}\ C]\!]$

$$[\![\mathbf{while}\ B\ \mathbf{do}\ C]\!] = f_{[\![B]\!],[\![C]\!]}([\![\mathbf{while}\ B\ \mathbf{do}\ C]\!])$$

where, for each $b : \mathit{State} \to \{\mathit{true}, \mathit{false}\}$ and $c : \mathit{State} \rightharpoonup \mathit{State}$, we define

$$f_{b,c} : (\mathit{State} \rightharpoonup \mathit{State}) \to (\mathit{State} \rightharpoonup \mathit{State})$$

as

$$f_{b,c} = \lambda w \in (\mathit{State} \rightharpoonup \mathit{State}).\, \lambda s \in \mathit{State}.\, \mathit{if}\left(b(s), w(c(s)), s\right).$$

- Why does $w = f_{[\![B]\!],[\![C]\!]}(w)$ have a solution?
- What if it has several solutions—which one do we take to be $[\![\mathbf{while}\ B\ \mathbf{do}\ C]\!]$?

Approximating $[\![\mathbf{while}\ B\ \mathbf{do}\ C]\!]$

Approximating $[\![\mathbf{while}\ B\ \mathbf{do}\ C]\!]$

$$
\begin{aligned}
& {f_{[\![B]\!],[\![C]\!]}}^{n}(\bot) \\
= \quad & \lambda s \in \mathit{State}. \\
& \begin{cases}
[\![C]\!]^{k}(s) & \text{if } \exists\, 0 \leq k < n.\, [\![B]\!]([\![C]\!]^{k}(s)) = \mathit{false} \\
& \text{and } \forall\, 0 \leq i < k.\, [\![B]\!]([\![C]\!]^{i}(s)) = \mathit{true} \\
\\
\uparrow & \text{if } \forall\, 0 \leq i < n.\, [\![B]\!]([\![C]\!]^{i}(s)) = \mathit{true}
\end{cases}
\end{aligned}
$$

the space of state transformers.

$$D \stackrel{\text{def}}{=} (State \rightharpoonup State)$$

PARTIAL ORDER $(P, \sqsubseteq)$ $\sqsubseteq \subseteq P \times P$

a relation

satisfies 3 properties

- **Partial order $\sqsubseteq$ on D:**

  $w \sqsubseteq w'$ iff for all $s \in State$, if w is defined at s then so is w' and moreover $w(s) = w'(s)$.

  iff the graph of w is included in the graph of w'.

- **Least element $\bot \in D$ w.r.t. $\sqsubseteq$:**

  $\bot$ = totally undefined partial function

  = partial function with empty graph

(satisfies $\bot \sqsubseteq w$, for all $w \in D$).

- $p \sqsubseteq p$
- $p \sqsubseteq q \;\&\; q \sqsubseteq r \Rightarrow p \sqsubseteq r$
- $p \sqsubseteq q \;\&\; q \sqsubseteq p \Rightarrow p = q$

Topic 2

Least Fixed Points

Thesis

All domains of computation are
partial orders with a least element.

For posets $(P, \sqsubseteq_P)$ and $(Q, \sqsubseteq_Q)$ a function $f : P \to Q$ is monotone if $p_1 \sqsubseteq p_2 \Rightarrow f(p_1) \sqsubseteq f(p_2)$

Thesis

All domains of computation are partial orders with a least element.

All computable functions are mononotic.

Example $\forall B, C$.

$f_{[\![B]\!],[\![C]\!]} : (State \to State) \to (State \to State)$ monotone.

A consequence of monotonicity

$[\![\text{while } B \text{ do } C]\!]_0 = \perp$

$[\![\text{while } B \text{ do } C]\!]_{n+1} = f_{[\![B]\!],[\![C]\!]}([\![\text{while } B \text{ do } C]\!]_n)$

We have:

$$[\![\text{while } B \text{ do } C]\!]_k \sqsubseteq [\![\text{while } B \text{ do } C]\!]_{k+1}$$

$(P, \sqsubseteq)$ poset with least element $\quad p \in P \quad f$ monotone $(P,\sqsubseteq) \to (P,\sqsubseteq)$

$\perp \sqsubseteq p \Rightarrow \perp \sqsubseteq f(\perp) \Rightarrow f(\perp) \sqsubseteq f f(\perp)$

$\Rightarrow f f \perp \sqsubseteq f^3(\perp)$

By ind. $f^i(\perp) \sqsubseteq f^j(\perp) \; \forall i \leq j$