

`[[while B do C]]`

$\llbracket \text{while } B \text{ do } C \rrbracket : \text{State} \rightarrow \text{State}$

// def

... $\llbracket B \rrbracket$... $\llbracket C \rrbracket$

sanity check such that

$\llbracket \text{while } B \text{ do } C \rrbracket(s)$

A fixed point of
a partial function

$f : S \rightarrow S$ is an element

$x \in S$ s.t. $f(x) = x$

= $\#(\llbracket B \rrbracket s,$

$\llbracket \text{while } B \text{ do } C \rrbracket(\llbracket C \rrbracket s),$

$s)$

What function is
 $\llbracket \text{while } B \text{ do } C \rrbracket$ a
fixed point of ?

$\boxed{\text{while } B \text{ do } C} = \lambda s. \boxed{\boxed{\text{if } (B)(s),}$
 $\underbrace{\boxed{\text{while } B \text{ do } C}}_{s}) (\bar{a}cys),$

$\lambda w: \text{State} \rightarrow \text{State}. \lambda s: \text{State}.$
 $\boxed{\boxed{\text{if } (B)(s), W(\boxed{C}(s)), s)}$

Fixed point property of [while B do C]

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

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

$$f_{b,c} : (\text{State} \rightarrow \text{State}) \rightarrow (\text{State} \rightarrow \text{State})$$

as

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

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

Approximating $\llbracket \text{while } B \text{ do } C \rrbracket$

domain $(\text{State} \rightarrow \text{State})$

contains the empty/completely undefined partial function: \perp (i.e. $\perp(s) \uparrow \forall s \in \text{State}$).

$$W_0 = \perp$$

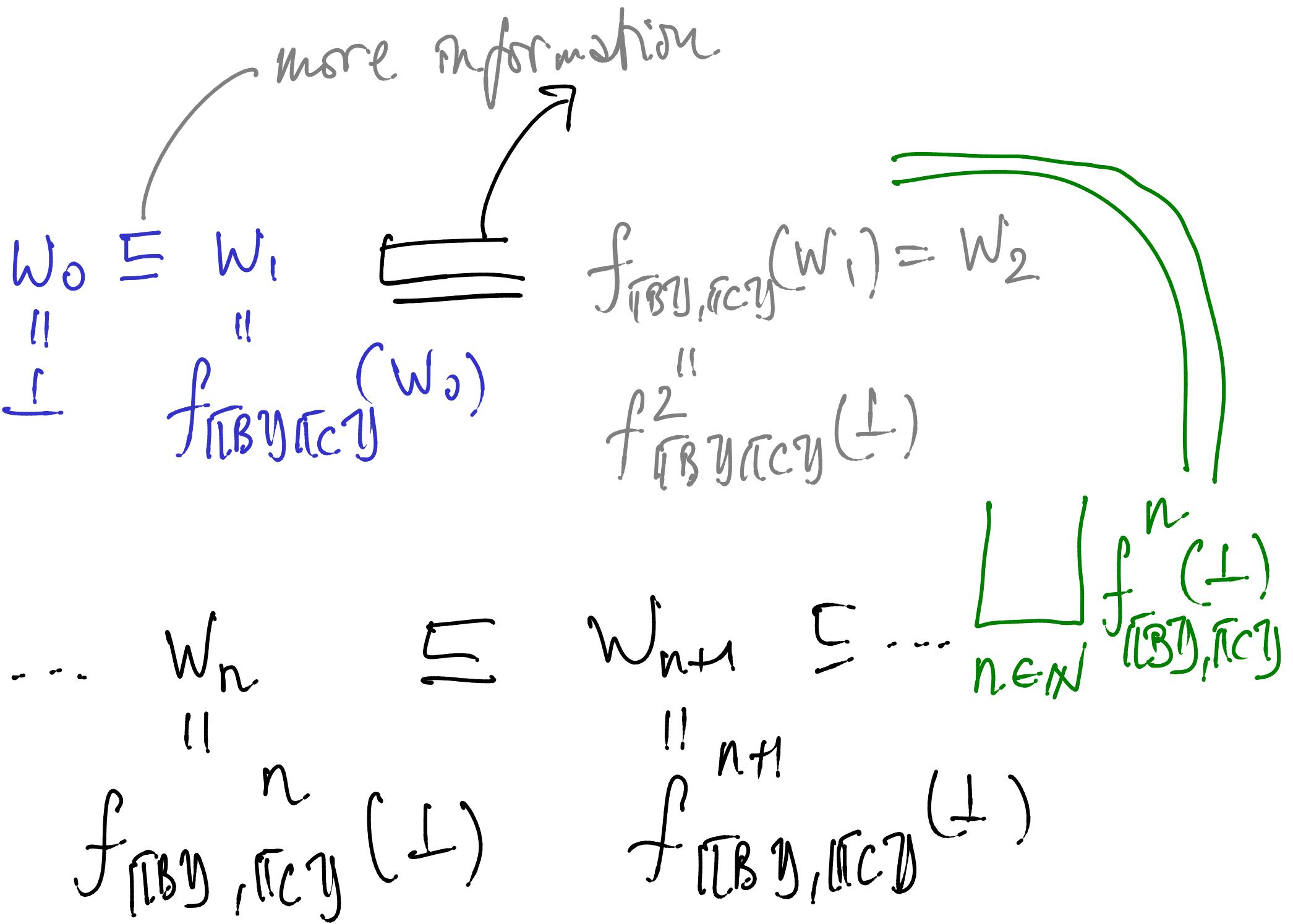
$$\begin{aligned} W_1 &= (\lambda w. \lambda s. \text{if } (\bar{A}Bys, W(\bar{C}ys), s))(\perp) \\ &= \lambda s. \text{if } (\bar{A}Bys, \perp(\bar{C}ys), s) \\ &= \lambda s. \left\{ \begin{array}{ll} \uparrow & \text{if } Bys = \text{true} \\ s & \text{otherwise} \end{array} \right. \end{aligned}$$

$$W_2 = (\lambda w. \lambda s. \eta (\bar{A}B)ys, w(\bar{C}ys), s)(w_1)$$

$$= \lambda s. \eta (\bar{A}B)ys, w_1(\bar{C}s), s$$

$$= \lambda s. \begin{cases} s & \bar{A}Bys = \text{false} \\ w_1(\bar{C}ys) & \text{otherwise} \end{cases}$$

$$= \lambda s. \begin{cases} s & \bar{A}Bys = \text{false} \\ \bar{C}ys & \bar{B}(\bar{A}B)ys = \text{false} \\ & \text{otherwise} \end{cases}$$



Approximating $\llbracket \text{while } B \text{ do } C \rrbracket$

$$f_{\llbracket B \rrbracket, \llbracket C \rrbracket}^n(\perp)$$

$= \lambda s \in State.$

$$\begin{cases} \llbracket C \rrbracket^k(s) & \text{if } \exists 0 \leq k < n. \llbracket B \rrbracket(\llbracket C \rrbracket^k(s)) = \text{false} \\ & \text{and } \forall 0 \leq i < k. \llbracket B \rrbracket(\llbracket C \rrbracket^i(s)) = \text{true} \\ \uparrow & \text{if } \forall 0 \leq i < n. \llbracket B \rrbracket(\llbracket C \rrbracket^i(s)) = \text{true} \end{cases}$$

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

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

$w \sqsubseteq w'$ iff for all $s \in \text{State}$, 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 $\perp \in D$ w.r.t. \sqsubseteq :**

\perp = totally undefined partial function
= partial function with empty graph

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

Topic 2

Least Fixed Points

$\perp \leq x \wedge x \in D$

information
order.

Thesis

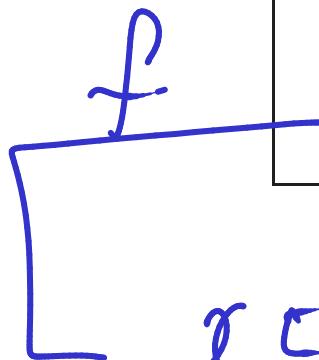
provides
no

information.

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

$\perp \leq f(\perp) \Rightarrow f(\perp) \leq f^2(\perp) \Rightarrow \dots \Rightarrow f^n(\perp) \leq f^{n+1}(\perp)$

All computable functions are
monotonic.



$x \leq y \Rightarrow f(x) \leq f(y)$

$\perp \leq f(\perp) \leq f^2(\perp) \leq \dots \leq f^n(\perp) \leq \dots$

fixed point of
 f ?

? $\bigcup_n f^n(\perp)$?