Semantics and Verification of Software

Lecture 5: Basic Fixpoint Theory

Thomas Noll

Lehrstuhl für Informatik 2
RWTH Aachen University
noll@cs.rwth-aachen.de

http://www-i2.informatik.rwth-aachen.de/i2/svsw/

Summer semester 2007

Outline

1. Repetition: Denotational Semantics
2. Characterization of fix(Φ)
3. Chain–Complete Partial Orders

Semantics of Statements II

Definition (Denotational semantics of statements)

The (denotational) semantic functional for statements,

$$\mathfrak{C}[\![.]\!] : Cmd \to (\Sigma \dashrightarrow \Sigma),$$

is given by:

$$\begin{aligned}
\mathfrak{C}[\![\texttt{skip}]\!] &:= \mathsf{id}_\Sigma \\
\mathfrak{C}[\![x \texttt{ := } a]\!]\sigma &:= \sigma[x \mapsto \mathfrak{A}[\![a]\!]\sigma] \\
\mathfrak{C}[\![c_1 \texttt{;} c_2]\!] &:= \mathfrak{C}[\![c_2]\!] \circ \mathfrak{C}[\![c_1]\!] \\
\mathfrak{C}[\![\texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2]\!] &:= \mathsf{cond}(\mathfrak{B}[\![b]\!], \mathfrak{C}[\![c_1]\!], \mathfrak{C}[\![c_2]\!]) \\
\mathfrak{C}[\![\texttt{while } b \texttt{ do } c]\!] &:= \mathsf{fix}(\Phi)
\end{aligned}$$

where $\Phi : (\Sigma \dashrightarrow \Sigma) \to (\Sigma \dashrightarrow \Sigma) : f \mapsto \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$

Why Fixpoints?

- Goal: preserve validity of equivalence
  $$\mathfrak{C}[\![\texttt{while}\ b\ \texttt{do}\ c]\!] = \mathfrak{C}[\![\texttt{if}\ b\ \texttt{then}\ (c\texttt{;while}\ b\ \texttt{do}\ c)\ \texttt{else}\ \texttt{skip}]\!]$$
- Using the known parts of Def. 4.8, we obtain:

$$\begin{aligned}
&\mathfrak{C}[\![\texttt{while}\ b\ \texttt{do}\ c]\!] \\
&= \mathfrak{C}[\![\texttt{if}\ b\ \texttt{then}\ (c\texttt{;while}\ b\ \texttt{do}\ c)\ \texttt{else}\ \texttt{skip}]\!] \\
&= \mathsf{cond}(\mathfrak{B}[\![b]\!], \mathfrak{C}[\![c\texttt{;while}\ b\ \texttt{do}\ c]\!], \mathfrak{C}[\![\texttt{skip}]\!]) \\
&= \mathsf{cond}(\mathfrak{B}[\![b]\!], \mathfrak{C}[\![\texttt{while}\ b\ \texttt{do}\ c]\!] \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)
\end{aligned}$$

- Abbreviating $f := \mathfrak{C}[\![\texttt{while}\ b\ \texttt{do}\ c]\!]$ this yields:
  $$f = \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$$
- Hence f must be a solution of this recursive equation
- Or: f must be a fixpoint of the mapping
  $$\Phi : (\Sigma \dashrightarrow \Sigma) \to (\Sigma \dashrightarrow \Sigma) : f \mapsto \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$$
  (since the equation can be stated as $f = \Phi(f)$)

Well–Definedness of Fixpoint Semantics

But: fixpoint property not sufficient to obtain a well–defined semantics

Existence: there does no need to exist any fixpoint. Examples:

1. $\phi_1 : \mathbb{N} \to \mathbb{N} : n \mapsto n + 1$ has no fixpoint
2. $\Phi_1 : (\Sigma \dashrightarrow \Sigma) \to (\Sigma \dashrightarrow \Sigma) : f \mapsto \begin{cases} g_1 & \text{if } f = g_2 \\ g_2 & \text{otherwise} \end{cases}$

   (where $g_1 \neq g_2$) has no fixpoint

Solution: in our setting, fixpoints always exist

Uniqueness: there might exist several fixpoints. Examples:

1. $\phi_2 : \mathbb{N} \to \mathbb{N} : n \mapsto n^3$ has fixpoints $\{0, 1\}$
2. every state transformation f is a fixpoint of

   $\Phi_2 : (\Sigma \dashrightarrow \Sigma) \to (\Sigma \dashrightarrow \Sigma) : f \mapsto f$

Solution: guaranteed by choosing a special fixpoint

Outline

1. Repetition: Denotational Semantics
2. Characterization of $\mathsf{fix}(\Phi)$
3. Chain–Complete Partial Orders

- Let $b \in BExp$ and $c \in Cmd$
- Let $\Phi(f) := \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$
- Let $f_0 : \Sigma \dashrightarrow \Sigma$ be a fixpoint of Φ, i.e., $\Phi(f_0) = f_0$
- Given some initial state $\sigma_0 \in \Sigma$, we will distinguish the following cases:
  1. loop `while` b `do` c terminates after n iterations ($n \in \mathbb{N}$)
  2. body c diverges in the nth iteration (since it contains a non–terminating `while` statement)
  3. loop `while` b `do` c itself diverges

Characterization of $\mathsf{fix}(\Phi)$ I

- Let $b \in \mathit{BExp}$ and $c \in \mathit{Cmd}$
- Let $\Phi(f) := \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$
- Let $f_0 : \Sigma \dashrightarrow \Sigma$ be a fixpoint of Φ, i.e., $\Phi(f_0) = f_0$
- Given some initial state $\sigma_0 \in \Sigma$, we will distinguish the following cases:
  1. loop `while` b `do` c terminates after n iterations ($n \in \mathbb{N}$)
  2. body c diverges in the nth iteration (since it contains a non–terminating `while` statement)
  3. loop `while` b `do` c itself diverges

Characterization of $\mathsf{fix}(\Phi)$ I

- Let $b \in \mathit{BExp}$ and $c \in \mathit{Cmd}$
- Let $\Phi(f) := \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$
- Let $f_0 : \Sigma \dashrightarrow \Sigma$ be a fixpoint of Φ, i.e., $\Phi(f_0) = f_0$
- Given some initial state $\sigma_0 \in \Sigma$, we will distinguish the following cases:
  1. loop `while` b `do` c terminates after n iterations ($n \in \mathbb{N}$)
  2. body c diverges in the nth iteration (since it contains a non–terminating `while` statement)
  3. loop `while` b `do` c itself diverges

Characterization of $\mathsf{fix}(\Phi)$ I

- Let $b \in \mathit{BExp}$ and $c \in \mathit{Cmd}$
- Let $\Phi(f) := \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$
- Let $f_0 : \Sigma \dashrightarrow \Sigma$ be a fixpoint of Φ, i.e., $\Phi(f_0) = f_0$
- Given some initial state $\sigma_0 \in \Sigma$, we will distinguish the following cases:
  1. loop `while` b `do` c terminates after n iterations ($n \in \mathbb{N}$)
  2. body c diverges in the nth iteration (since it contains a non–terminating `while` statement)
  3. loop `while` b `do` c itself diverges

Case 1: Termination of Loop

- Loop `while` b `do` c terminates after n iterations $(n \in \mathbb{N})$
- Formally: there exist $\sigma_1, \ldots, \sigma_n \in \Sigma$ such that

$$\mathfrak{B}[\![b]\!]\sigma_i = \begin{cases} \mathsf{true} & \text{if } 0 \le i < n \\ \mathsf{false} & \text{if } i = n \end{cases} \quad \text{and}$$

$$\mathfrak{C}[\![c]\!]\sigma_i = \sigma_{i+1} \quad \text{for every } 0 \le i < n$$

- Now the definition
$\Phi(f) := \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$ implies, for every $0 \le i < n$,

$$\begin{aligned} \Phi(f_0)(\sigma_i) &= (f_0 \circ \mathfrak{C}[\![c]\!])(\sigma_i) && \text{since } \mathfrak{B}[\![b]\!]\sigma_i = \mathsf{true} \\ &= f_0(\sigma_{i+1}) && \text{and} \\ \Phi(f_0)(\sigma_n) &= \sigma_n && \text{since } \mathfrak{B}[\![b]\!]\sigma_n = \mathsf{false} \end{aligned}$$

- Since $\Phi(f_0) = f_0$ it follows that

$$f_0(\sigma_i) = \begin{cases} f_0(\sigma_{i+1}) & \text{if } 0 \le i < n \\ \sigma_n & \text{if } i = n \end{cases}$$

and hence

$$f_0(\sigma_0) = f_0(\sigma_1) = \ldots f_0(\sigma_n) = \sigma_n$$

$\Longrightarrow$ All fixpoints coincide on σ_0!

Case 1: Termination of Loop

- Loop `while` b `do` c terminates after n iterations ($n \in \mathbb{N}$)
- Formally: there exist $\sigma_1, \ldots, \sigma_n \in \Sigma$ such that

$$\mathfrak{B}[\![b]\!]\sigma_i = \begin{cases} \mathsf{true} & \text{if } 0 \leq i < n \\ \mathsf{false} & \text{if } i = n \end{cases} \quad \text{and}$$

$$\mathfrak{C}[\![c]\!]\sigma_i = \sigma_{i+1} \quad \text{for every } 0 \leq i < n$$

- Now the definition

  $\Phi(f) := \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$ implies, for every $0 \leq i < n$,

$$\begin{aligned} \Phi(f_0)(\sigma_i) &= (f_0 \circ \mathfrak{C}[\![c]\!])(\sigma_i) && \text{since } \mathfrak{B}[\![b]\!]\sigma_i = \mathsf{true} \\ &= f_0(\sigma_{i+1}) && \text{and} \\ \Phi(f_0)(\sigma_n) &= \sigma_n && \text{since } \mathfrak{B}[\![b]\!]\sigma_n = \mathsf{false} \end{aligned}$$

- Since $\Phi(f_0) = f_0$ it follows that

$$f_0(\sigma_i) = \begin{cases} f_0(\sigma_{i+1}) & \text{if } 0 \leq i < n \\ \sigma_n & \text{if } i = n \end{cases}$$

  and hence

$$f_0(\sigma_0) = f_0(\sigma_1) = \ldots f_0(\sigma_n) = \sigma_n$$

$\implies$ All fixpoints coincide on σ_0!

Case 1: Termination of Loop

- Loop `while` b `do` c terminates after n iterations ($n \in \mathbb{N}$)
- Formally: there exist $\sigma_1, \ldots, \sigma_n \in \Sigma$ such that

$$\mathfrak{B}[\![b]\!]\sigma_i = \begin{cases} \mathsf{true} & \text{if } 0 \le i < n \\ \mathsf{false} & \text{if } i = n \end{cases} \quad \text{and}$$

$$\mathfrak{C}[\![c]\!]\sigma_i = \sigma_{i+1} \quad \text{for every } 0 \le i < n$$

- Now the definition
  $\Phi(f) := \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$ implies, for every $0 \le i < n$,

$$\begin{aligned} \Phi(f_0)(\sigma_i) &= (f_0 \circ \mathfrak{C}[\![c]\!])(\sigma_i) && \text{since } \mathfrak{B}[\![b]\!]\sigma_i = \mathsf{true} \\ &= f_0(\sigma_{i+1}) && \text{and} \\ \Phi(f_0)(\sigma_n) &= \sigma_n && \text{since } \mathfrak{B}[\![b]\!]\sigma_n = \mathsf{false} \end{aligned}$$

- Since $\Phi(f_0) = f_0$ it follows that

$$f_0(\sigma_i) = \begin{cases} f_0(\sigma_{i+1}) & \text{if } 0 \le i < n \\ \sigma_n & \text{if } i = n \end{cases}$$

  and hence

$$f_0(\sigma_0) = f_0(\sigma_1) = \ldots f_0(\sigma_n) = \sigma_n$$

$\Longrightarrow$ All fixpoints coincide on σ_0!

Case 1: Termination of Loop

- Loop `while` b `do` c terminates after n iterations ($n \in \mathbb{N}$)
- Formally: there exist $\sigma_1, \ldots, \sigma_n \in \Sigma$ such that

$$\mathfrak{B}[\![b]\!]\sigma_i = \begin{cases} \mathsf{true} & \text{if } 0 \leq i < n \\ \mathsf{false} & \text{if } i = n \end{cases} \quad \text{and}$$

$$\mathfrak{C}[\![c]\!]\sigma_i = \sigma_{i+1} \quad \text{for every } 0 \leq i < n$$

- Now the definition
  $\Phi(f) := \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$ implies, for every $0 \leq i < n$,

$$\begin{aligned} \Phi(f_0)(\sigma_i) &= (f_0 \circ \mathfrak{C}[\![c]\!])(\sigma_i) && \text{since } \mathfrak{B}[\![b]\!]\sigma_i = \mathsf{true} \\ &= f_0(\sigma_{i+1}) && \text{and} \\ \Phi(f_0)(\sigma_n) &= \sigma_n && \text{since } \mathfrak{B}[\![b]\!]\sigma_n = \mathsf{false} \end{aligned}$$

- Since $\Phi(f_0) = f_0$ it follows that

$$f_0(\sigma_i) = \begin{cases} f_0(\sigma_{i+1}) & \text{if } 0 \leq i < n \\ \sigma_n & \text{if } i = n \end{cases}$$

  and hence

$$f_0(\sigma_0) = f_0(\sigma_1) = \ldots f_0(\sigma_n) = \sigma_n$$

$\Longrightarrow$ All fixpoints coincide on σ_0!

Case 1: Termination of Loop

- Loop `while` b `do` c terminates after n iterations ($n \in \mathbb{N}$)
- Formally: there exist $\sigma_1, \ldots, \sigma_n \in \Sigma$ such that

$$\mathfrak{B}[\![b]\!]\sigma_i = \begin{cases} \mathsf{true} & \text{if } 0 \leq i < n \\ \mathsf{false} & \text{if } i = n \end{cases} \quad \text{and}$$

$$\mathfrak{C}[\![c]\!]\sigma_i = \sigma_{i+1} \quad \text{for every } 0 \leq i < n$$

- Now the definition
  $\Phi(f) := \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$ implies, for every $0 \leq i < n$,

$$\begin{aligned} \Phi(f_0)(\sigma_i) &= (f_0 \circ \mathfrak{C}[\![c]\!])(\sigma_i) && \text{since } \mathfrak{B}[\![b]\!]\sigma_i = \mathsf{true} \\ &= f_0(\sigma_{i+1}) && \text{and} \\ \Phi(f_0)(\sigma_n) &= \sigma_n && \text{since } \mathfrak{B}[\![b]\!]\sigma_n = \mathsf{false} \end{aligned}$$

- Since $\Phi(f_0) = f_0$ it follows that

$$f_0(\sigma_i) = \begin{cases} f_0(\sigma_{i+1}) & \text{if } 0 \leq i < n \\ \sigma_n & \text{if } i = n \end{cases}$$

  and hence

$$f_0(\sigma_0) = f_0(\sigma_1) = \ldots f_0(\sigma_n) = \sigma_n$$

$\Longrightarrow$ All fixpoints coincide on σ_0!

Case 2: Divergence of Body

- Body c diverges in the nth iteration (since it contains a non–terminating `while` statement)
- Formally: there exist $\sigma_1, \ldots, \sigma_{n-1} \in \Sigma$ such that

$$\mathfrak{B}[\![b]\!]\sigma_i = \mathsf{true} \qquad \text{for every } 0 \leq i < n \text{ and}$$

$$\mathfrak{C}[\![c]\!]\sigma_i = \begin{cases} \sigma_{i+1} & \text{if } 0 \leq i \leq n-2 \\ \text{undefined} & \text{if } i = n-1 \end{cases}$$

- Just as in the previous case (setting $\sigma_n :=$ undefined) it follows that

$$f_0(\sigma_0) = \text{undefined}$$

$\Longrightarrow$ Again all fixpoints coincide on σ_0!

- Body c diverges in the nth iteration (since it contains a non–terminating `while` statement)
- Formally: there exist $\sigma_1, \ldots, \sigma_{n-1} \in \Sigma$ such that

$$\mathfrak{B}[\![b]\!]\sigma_i = \mathsf{true} \qquad \text{for every } 0 \le i < n \text{ and}$$

$$\mathfrak{C}[\![c]\!]\sigma_i = \begin{cases} \sigma_{i+1} & \text{if } 0 \le i \le n-2 \\ \text{undefined} & \text{if } i = n-1 \end{cases}$$

- Just as in the previous case (setting $\sigma_n :=$ undefined) it follows that

$$f_0(\sigma_0) = \text{undefined}$$

$\Longrightarrow$ Again all fixpoints coincide on σ_0!

Case 2: Divergence of Body

- Body c diverges in the nth iteration (since it contains a non–terminating `while` statement)
- Formally: there exist $\sigma_1, \ldots, \sigma_{n-1} \in \Sigma$ such that

$$\mathfrak{B}[\![b]\!]\sigma_i = \mathsf{true} \qquad \text{for every } 0 \leq i < n \text{ and}$$

$$\mathfrak{C}[\![c]\!]\sigma_i = \begin{cases} \sigma_{i+1} & \text{if } 0 \leq i \leq n-2 \\ \text{undefined} & \text{if } i = n-1 \end{cases}$$

- Just as in the previous case (setting $\sigma_n :=$ undefined) it follows that

$$f_0(\sigma_0) = \text{undefined}$$

$\Longrightarrow$ Again all fixpoints coincide on σ_0!

Case 2: Divergence of Body

- Body c diverges in the nth iteration (since it contains a non–terminating `while` statement)
- Formally: there exist $\sigma_1, \ldots, \sigma_{n-1} \in \Sigma$ such that

$$\mathfrak{B}[\![b]\!]\sigma_i = \mathsf{true} \qquad \text{for every } 0 \leq i < n \text{ and}$$
$$\mathfrak{C}[\![c]\!]\sigma_i = \begin{cases} \sigma_{i+1} & \text{if } 0 \leq i \leq n-2 \\ \text{undefined} & \text{if } i = n-1 \end{cases}$$

- Just as in the previous case (setting $\sigma_n :=$ undefined) it follows that

$$f_0(\sigma_0) = \text{undefined}$$

$\Longrightarrow$ Again all fixpoints coincide on σ_0!

- Loop `while` b `do` c diverges
- Formally: there exist $\sigma_1, \sigma_2, \ldots \in \Sigma$ such that

$$\begin{aligned} \mathfrak{B}[\![b]\!]\sigma_i &= \mathsf{true} && \text{and} \\ \mathfrak{C}[\![c]\!]\sigma_i &= \sigma_{i+1} && \text{for every } i \in \mathbb{N} \end{aligned}$$

- Here only derivable:

$$f_0(\sigma_0) = f_0(\sigma_i) \quad \text{for every } i \in \mathbb{N}$$

$\Longrightarrow$ Value of $f_0(\sigma_0)$ not determined!

- Loop `while` b `do` c diverges
- Formally: there exist $\sigma_1, \sigma_2, \ldots \in \Sigma$ such that

$$\begin{array}{rcll} \mathfrak{B}[\![b]\!]\sigma_i &=& \mathsf{true} & \text{and} \\ \mathfrak{C}[\![c]\!]\sigma_i &=& \sigma_{i+1} & \text{for every } i \in \mathbb{N} \end{array}$$

- Here only derivable:

$$f_0(\sigma_0) = f_0(\sigma_i) \quad \text{for every } i \in \mathbb{N}$$

$\Longrightarrow$ Value of $f_0(\sigma_0)$ not determined!

Case 3: Divergence of Loop

- Loop `while` b `do` c diverges
- Formally: there exist $\sigma_1, \sigma_2, \ldots \in \Sigma$ such that

$$\begin{aligned} \mathfrak{B}[\![b]\!]\sigma_i &= \mathsf{true} && \text{and} \\ \mathfrak{C}[\![c]\!]\sigma_i &= \sigma_{i+1} && \text{for every } i \in \mathbb{N} \end{aligned}$$

- Here only derivable:

$$f_0(\sigma_0) = f_0(\sigma_i) \quad \text{for every } i \in \mathbb{N}$$

$\Longrightarrow$ Value of $f_0(\sigma_0)$ not determined!

Case 3: Divergence of Loop

- Loop `while` b `do` c diverges
- Formally: there exist $\sigma_1, \sigma_2, \ldots \in \Sigma$ such that

$$\begin{aligned} \mathfrak{B}[\![b]\!]\sigma_i &= \mathsf{true} && \text{and} \\ \mathfrak{C}[\![c]\!]\sigma_i &= \sigma_{i+1} && \text{for every } i \in \mathbb{N} \end{aligned}$$

- Here only derivable:

$$f_0(\sigma_0) = f_0(\sigma_i) \quad \text{for every } i \in \mathbb{N}$$

$\Longrightarrow$ Value of $f_0(\sigma_0)$ not determined!

Summary

For $\Phi(f_0) = f_0$ and initial state $\sigma_0 \in \Sigma$, case distinction yields:

1. Loop `while` b `do` c terminates after n iterations ($n \in \mathbb{N}$)
   $\implies f_0(\sigma_0) = \sigma_n$
2. Body c diverges in the nth iteration
   $\implies f_0(\sigma_0) = \text{undefined}$
3. Loop `while` b `do` c diverges
   $\implies$ no condition on f_0 (only $f_0(\sigma_0) = f_0(\sigma_i)$ for every $i \in \mathbb{N}$)

- Not surprising since, e.g., `while true do skip` yields for every $f : \Sigma \dashrightarrow \Sigma$

$$\Phi(f) = \mathsf{cond}(\mathfrak{B}[\![\mathsf{true}]\!], f \circ \mathfrak{C}[\![\mathsf{skip}]\!], \mathsf{id}_\Sigma) = f$$

- On the other hand, our operational understanding requires, for every $\sigma_0 \in \Sigma$,

$$\mathfrak{C}[\![\texttt{while true do skip}]\!]\sigma_0 = \text{undefined}$$

Conclusion

$\mathsf{fix}(\Phi)$ is the least defined fixpoint of Φ.

Summary

For $\Phi(f_0) = f_0$ and initial state $\sigma_0 \in \Sigma$, case distinction yields:

1. Loop `while` b `do` c terminates after n iterations ($n \in \mathbb{N}$)
   $\implies f_0(\sigma_0) = \sigma_n$
2. Body c diverges in the nth iteration
   $\implies f_0(\sigma_0) = \text{undefined}$
3. Loop `while` b `do` c diverges
   $\implies$ no condition on f_0 (only $f_0(\sigma_0) = f_0(\sigma_i)$ for every $i \in \mathbb{N}$)

- Not surprising since, e.g., `while true do skip` yields for every $f : \Sigma \dashrightarrow \Sigma$

$$\Phi(f) = \mathsf{cond}(\mathfrak{B}[\![\mathsf{true}]\!], f \circ \mathfrak{C}[\![\mathtt{skip}]\!], \mathsf{id}_\Sigma) = f$$

- On the other hand, our operational understanding requires, for every $\sigma_0 \in \Sigma$,

$$\mathfrak{C}[\![\mathtt{while\ true\ do\ skip}]\!]\sigma_0 = \text{undefined}$$

Conclusion

$\mathrm{fix}(\Phi)$ is the least defined fixpoint of Φ.

Summary

For $\Phi(f_0) = f_0$ and initial state $\sigma_0 \in \Sigma$, case distinction yields:

1. Loop `while` b `do` c terminates after n iterations ($n \in \mathbb{N}$)
   $\implies f_0(\sigma_0) = \sigma_n$
2. Body c diverges in the nth iteration
   $\implies f_0(\sigma_0) = \text{undefined}$
3. Loop `while` b `do` c diverges
   $\implies$ no condition on f_0 (only $f_0(\sigma_0) = f_0(\sigma_i)$ for every $i \in \mathbb{N}$)

- Not surprising since, e.g., `while true do skip` yields for every $f : \Sigma \dashrightarrow \Sigma$

$$\Phi(f) = \mathsf{cond}(\mathfrak{B}[\![\mathsf{true}]\!], f \circ \mathfrak{C}[\![\mathtt{skip}]\!], \mathsf{id}_\Sigma) = f$$

- On the other hand, our operational understanding requires, for every $\sigma_0 \in \Sigma$,

$$\mathfrak{C}[\![\mathtt{while\ true\ do\ skip}]\!]\sigma_0 = \text{undefined}$$

Conclusion

$\mathsf{fix}(\Phi)$ is the least defined fixpoint of Φ.

Summary

For $\Phi(f_0) = f_0$ and initial state $\sigma_0 \in \Sigma$, case distinction yields:

1. Loop `while` b `do` c terminates after n iterations ($n \in \mathbb{N}$)
   $\implies f_0(\sigma_0) = \sigma_n$
2. Body c diverges in the nth iteration
   $\implies f_0(\sigma_0) = \text{undefined}$
3. Loop `while` b `do` c diverges
   $\implies$ no condition on f_0 (only $f_0(\sigma_0) = f_0(\sigma_i)$ for every $i \in \mathbb{N}$)

- Not surprising since, e.g., `while true do skip` yields for every $f : \Sigma \dashrightarrow \Sigma$

$$\Phi(f) = \mathsf{cond}(\mathfrak{B}[\![\mathsf{true}]\!], f \circ \mathfrak{C}[\![\mathtt{skip}]\!], \mathsf{id}_\Sigma) = f$$

- On the other hand, our operational understanding requires, for every $\sigma_0 \in \Sigma$,

$$\mathfrak{C}[\![\mathtt{while\ true\ do\ skip}]\!]\sigma_0 = \text{undefined}$$

Conclusion

$\mathsf{fix}(\Phi)$ is the least defined fixpoint of Φ.

Making it Precise I

To use fixpoint theory, the notion of "least defined" has to be made precise.

- Given $f, g : \Sigma \dashrightarrow \Sigma$, let

$$f \sqsubseteq g \iff \text{for every } \sigma, \sigma' \in \Sigma : f(\sigma) = \sigma' \implies g(\sigma) = \sigma'$$

  (g is "at least as defined" as f)

- Equivalent to requiring

$$\mathsf{graph}(f) \subseteq \mathsf{graph}(g)$$

  where

$$\mathsf{graph}(h) := \{(\sigma, \sigma') \mid \sigma \in \Sigma, \sigma' = h(\sigma) \text{ defined}\} \subseteq \Sigma \times \Sigma$$

  for every $h : \Sigma \dashrightarrow \Sigma$

Making it Precise I

To use fixpoint theory, the notion of "least defined" has to be made precise.

- Given $f, g : \Sigma \dashrightarrow \Sigma$, let

$$f \sqsubseteq g \iff \text{for every } \sigma, \sigma' \in \Sigma : f(\sigma) = \sigma' \implies g(\sigma) = \sigma'$$

  (g is "at least as defined" as f)

- Equivalent to requiring

$$\mathsf{graph}(f) \subseteq \mathsf{graph}(g)$$

  where

$$\mathsf{graph}(h) := \{(\sigma, \sigma') \mid \sigma \in \Sigma, \sigma' = h(\sigma) \text{ defined}\} \subseteq \Sigma \times \Sigma$$

  for every $h : \Sigma \dashrightarrow \Sigma$

Making it Precise II

Example 5.1

Let $x \in Var$ be fixed, and let $f_0, f_1, f_2, f_3 : \Sigma \dashrightarrow \Sigma$ be given by

$$
\begin{aligned}
f_0(\sigma) &:= \text{undefined}\\
f_1(\sigma) &:= \begin{cases} \sigma & \text{if } \sigma(x) \text{ even}\\ \text{undefined} & \text{otherwise}\end{cases}\\
f_2(\sigma) &:= \begin{cases} \sigma & \text{if } \sigma(x) \text{ odd}\\ \text{undefined} & \text{otherwise}\end{cases}\\
f_3(\sigma) &:= \sigma
\end{aligned}
$$

This implies $f_0 \sqsubseteq f_1 \sqsubseteq f_3$, $f_0 \sqsubseteq f_2 \sqsubseteq f_3$, $f_1 \not\sqsubseteq f_2$, and $f_2 \not\sqsubseteq f_1$

Making it Precise II

Example 5.1

Let $x \in Var$ be fixed, and let $f_0, f_1, f_2, f_3 : \Sigma \dashrightarrow \Sigma$ be given by

$$\begin{aligned} f_0(\sigma) &:= \text{undefined} \\ f_1(\sigma) &:= \begin{cases} \sigma & \text{if } \sigma(x) \text{ even} \\ \text{undefined} & \text{otherwise} \end{cases} \\ f_2(\sigma) &:= \begin{cases} \sigma & \text{if } \sigma(x) \text{ odd} \\ \text{undefined} & \text{otherwise} \end{cases} \\ f_3(\sigma) &:= \sigma \end{aligned}$$

This implies $f_0 \sqsubseteq f_1 \sqsubseteq f_3$, $f_0 \sqsubseteq f_2 \sqsubseteq f_3$, $f_1 \not\sqsubseteq f_2$, and $f_2 \not\sqsubseteq f_1$

Now $\mathsf{fix}(\Phi)$ can be characterized by:

- $\mathsf{fix}(\Phi)$ is a fixpoint of Φ, i.e.,

$$\Phi(\mathsf{fix}(\Phi)) = \mathsf{fix}(\Phi)$$

- $\mathsf{fix}(\Phi)$ is minimal with respect to $\sqsubseteq$, i.e., for every $f_0 : \Sigma \dashrightarrow \Sigma$ such that $\Phi(f_0) = f_0$,

$$\mathsf{fix}(\Phi) \sqsubseteq f_0$$

Example 5.2

For `while true do skip` we obtain for every $f : \Sigma \dashrightarrow \Sigma$:

$$\Phi(f) = \mathsf{cond}(\mathfrak{B}[\![\mathsf{true}]\!], f \circ \mathfrak{C}[\![\mathtt{skip}]\!], \mathsf{id}_\Sigma) = f$$

$\Longrightarrow$ $\mathsf{fix}(\Phi) = f_\emptyset$ where $f_\emptyset(\sigma) :=$ undefined for every $\sigma \in \Sigma$ (that is, $\mathsf{graph}(f_\emptyset) = \emptyset$)

Characterization of fix(Φ) II

Now $\mathsf{fix}(\Phi)$ can be characterized by:

- $\mathsf{fix}(\Phi)$ is a fixpoint of Φ, i.e.,

$$\Phi(\mathsf{fix}(\Phi)) = \mathsf{fix}(\Phi)$$

- $\mathsf{fix}(\Phi)$ is minimal with respect to $\sqsubseteq$, i.e., for every $f_0 : \Sigma \dashrightarrow \Sigma$ such that $\Phi(f_0) = f_0$,

$$\mathsf{fix}(\Phi) \sqsubseteq f_0$$

Example 5.2

For `while true do skip` we obtain for every $f : \Sigma \dashrightarrow \Sigma$:

$$\Phi(f) = \mathsf{cond}(\mathfrak{B}[\![\mathsf{true}]\!], f \circ \mathfrak{C}[\![\mathtt{skip}]\!], \mathsf{id}_\Sigma) = f$$

$\Longrightarrow$ $\mathsf{fix}(\Phi) = f_\emptyset$ where $f_\emptyset(\sigma) :=$ undefined for every $\sigma \in \Sigma$
(that is, $\mathsf{graph}(f_\emptyset) = \emptyset$)

Characterization of fix(Φ) II

Now fix(Φ) can be characterized by:

- fix(Φ) is a fixpoint of Φ, i.e.,

$$\Phi(\mathsf{fix}(\Phi)) = \mathsf{fix}(\Phi)$$

- fix(Φ) is minimal with respect to $\sqsubseteq$, i.e., for every $f_0 : \Sigma \dashrightarrow \Sigma$ such that $\Phi(f_0) = f_0$,

$$\mathsf{fix}(\Phi) \sqsubseteq f_0$$

Example 5.2

For `while true do skip` we obtain for every $f : \Sigma \dashrightarrow \Sigma$:

$$\Phi(f) = \mathsf{cond}(\mathfrak{B}[\![\mathsf{true}]\!], f \circ \mathfrak{C}[\![\mathtt{skip}]\!], \mathsf{id}_\Sigma) = f$$

$\Longrightarrow$ $\mathsf{fix}(\Phi) = f_\emptyset$ where $f_\emptyset(\sigma) :=$ undefined for every $\sigma \in \Sigma$
(that is, $\mathsf{graph}(f_\emptyset) = \emptyset$)

Characterization of $\mathsf{fix}(\Phi)$ III

Goals:

- Prove existence of $\mathsf{fix}(\Phi)$ for $\Phi(f) = \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$
- Show how it can be "computed" (more exactly: approximated)

Sufficient conditions:

on domain $\Sigma \dashrightarrow \Sigma$: chain–complete partial order

on function Φ: continuity

Goals:

- Prove existence of $\mathsf{fix}(\Phi)$ for $\Phi(f) = \mathsf{cond}(\mathfrak{B}[\![b]\!], f \circ \mathfrak{C}[\![c]\!], \mathsf{id}_\Sigma)$
- Show how it can be "computed" (more exactly: approximated)

Sufficient conditions:

on domain $\Sigma \dashrightarrow \Sigma$: chain–complete partial order

on function Φ: continuity

Outline

1. Repetition: Denotational Semantics
2. Characterization of $\mathrm{fix}(\Phi)$
3. Chain–Complete Partial Orders

Partial Orders

Definition 5.3 (Partial order)

A partial order (PO) $(D, \sqsubseteq)$ consists of a set D, called domain, and of a relation $\sqsubseteq \subseteq D \times D$ such that, for every $d_1, d_2, d_3 \in D$,

- reflexivity: $d_1 \sqsubseteq d_1$
- transitivity: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_3 \implies d_1 \sqsubseteq d_3$
- antisymmetry: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_1 \implies d_1 = d_2$

It is called total if, in addition, always $d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$.

Example 5.4

1. $(\mathbb{N}, \leq)$ is a total partial order
2. $(2^{\mathbb{N}}, \subseteq)$ is a (non–total) partial order
3. $(\mathbb{N}, <)$ is not a partial order (since not reflexive)

Partial Orders

Definition 5.3 (Partial order)

A partial order (PO) $(D, \sqsubseteq)$ consists of a set D, called domain, and of a relation $\sqsubseteq \subseteq D \times D$ such that, for every $d_1, d_2, d_3 \in D$,

reflexivity: $d_1 \sqsubseteq d_1$

transitivity: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_3 \implies d_1 \sqsubseteq d_3$

antisymmetry: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_1 \implies d_1 = d_2$

It is called total if, in addition, always $d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$.

Example 5.4

1. $(\mathbb{N}, \leq)$ is a total partial order
2. $(2^{\mathbb{N}}, \subseteq)$ is a (non–total) partial order
3. $(\mathbb{N}, <)$ is not a partial order (since not reflexive)

Partial Orders

Definition 5.3 (Partial order)

A partial order (PO) $(D, \sqsubseteq)$ consists of a set D, called domain, and of a relation $\sqsubseteq \subseteq D \times D$ such that, for every $d_1, d_2, d_3 \in D$,

reflexivity: $d_1 \sqsubseteq d_1$

transitivity: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_3 \implies d_1 \sqsubseteq d_3$

antisymmetry: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_1 \implies d_1 = d_2$

It is called total if, in addition, always $d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$.

Example 5.4

1. $(\mathbb{N}, \leq)$ is a total partial order
2. $(2^{\mathbb{N}}, \subseteq)$ is a (non–total) partial order
3. $(\mathbb{N}, <)$ is not a partial order (since not reflexive)

Partial Orders

Definition 5.3 (Partial order)

A partial order (PO) $(D, \sqsubseteq)$ consists of a set D, called domain, and of a relation $\sqsubseteq \subseteq D \times D$ such that, for every $d_1, d_2, d_3 \in D$,

reflexivity: $d_1 \sqsubseteq d_1$

transitivity: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_3 \implies d_1 \sqsubseteq d_3$

antisymmetry: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_1 \implies d_1 = d_2$

It is called total if, in addition, always $d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$.

Example 5.4

1. $(\mathbb{N}, \leq)$ is a total partial order
2. $(2^{\mathbb{N}}, \subseteq)$ is a (non–total) partial order
3. $(\mathbb{N}, <)$ is not a partial order (since not reflexive)

Partial Orders

Definition 5.3 (Partial order)

A partial order (PO) $(D, \sqsubseteq)$ consists of a set D, called domain, and of a relation $\sqsubseteq \subseteq D \times D$ such that, for every $d_1, d_2, d_3 \in D$,

reflexivity: $d_1 \sqsubseteq d_1$

transitivity: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_3 \implies d_1 \sqsubseteq d_3$

antisymmetry: $d_1 \sqsubseteq d_2$ and $d_2 \sqsubseteq d_1 \implies d_1 = d_2$

It is called total if, in addition, always $d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$.

Example 5.4

1. $(\mathbb{N}, \leq)$ is a total partial order
2. $(2^{\mathbb{N}}, \subseteq)$ is a (non–total) partial order
3. $(\mathbb{N}, <)$ is not a partial order (since not reflexive)

Application to fix(Φ) I

Lemma 5.5

$(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$ *is a partial order.*

Proof.

see exercise □

Application to fix(Φ) I

Lemma 5.5

$(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$ *is a partial order.*

Proof.

see exercise □

Definition 5.6 (Chain, (least) upper bound)

Let $(D, \sqsubseteq)$ be a partial order and $S \subseteq D$.

1. S is called a chain in D if, for every $s_1, s_2 \in S$,
$$s_1 \sqsubseteq s_2 \text{ or } s_2 \sqsubseteq s_1$$
(that is, S is a totally ordered subset of D).
2. An element $d \in D$ is called an upper bound of S if $s \sqsubseteq d$ for every $s \in S$ (notation: $S \sqsubseteq d$).
3. An upper bound d of S is called least upper bound (LUB) or supremum of S if $d \sqsubseteq d'$ for every upper bound d' of S (notation: $d = \bigsqcup S$).

Example 5.7

1. Every subset $S \subseteq \mathbb{N}$ is a chain in $(\mathbb{N}, \leq)$. It has a LUB (its greatest element) iff it is finite.
2. $\{\emptyset, \{0\}, \{0, 1\}, \ldots\}$ is a chain in $(2^{\mathbb{N}}, \subseteq)$ with LUB $\mathbb{N}$.

Chains and Least Upper Bounds

Definition 5.6 (Chain, (least) upper bound)

Let $(D, \sqsubseteq)$ be a partial order and $S \subseteq D$.

1. S is called a chain in D if, for every $s_1, s_2 \in S$,
$$s_1 \sqsubseteq s_2 \text{ or } s_2 \sqsubseteq s_1$$
(that is, S is a totally ordered subset of D).
2. An element $d \in D$ is called an upper bound of S if $s \sqsubseteq d$ for every $s \in S$ (notation: $S \sqsubseteq d$).
3. An upper bound d of S is called least upper bound (LUB) or supremum of S if $d \sqsubseteq d'$ for every upper bound d' of S (notation: $d = \bigsqcup S$).

Example 5.7

1. Every subset $S \subseteq \mathbb{N}$ is a chain in $(\mathbb{N}, \leq)$.
It has a LUB (its greatest element) iff it is finite.
2. $\{\emptyset, \{0\}, \{0, 1\}, \ldots\}$ is a chain in $(2^{\mathbb{N}}, \subseteq)$ with LUB $\mathbb{N}$.

Chains and Least Upper Bounds

Definition 5.6 (Chain, (least) upper bound)

Let $(D, \sqsubseteq)$ be a partial order and $S \subseteq D$.

1. S is called a chain in D if, for every $s_1, s_2 \in S$,
$$s_1 \sqsubseteq s_2 \text{ or } s_2 \sqsubseteq s_1$$
(that is, S is a totally ordered subset of D).
2. An element $d \in D$ is called an upper bound of S if $s \sqsubseteq d$ for every $s \in S$ (notation: $S \sqsubseteq d$).
3. An upper bound d of S is called least upper bound (LUB) or supremum of S if $d \sqsubseteq d'$ for every upper bound d' of S (notation: $d = \bigsqcup S$).

Example 5.7

1. Every subset $S \subseteq \mathbb{N}$ is a chain in $(\mathbb{N}, \leq)$.
It has a LUB (its greatest element) iff it is finite.
2. $\{\emptyset, \{0\}, \{0, 1\}, \ldots\}$ is a chain in $(2^{\mathbb{N}}, \subseteq)$ with LUB $\mathbb{N}$.

Chains and Least Upper Bounds

Definition 5.6 (Chain, (least) upper bound)

Let $(D, \sqsubseteq)$ be a partial order and $S \subseteq D$.

1. S is called a chain in D if, for every $s_1, s_2 \in S$,
$$s_1 \sqsubseteq s_2 \text{ or } s_2 \sqsubseteq s_1$$
(that is, S is a totally ordered subset of D).
2. An element $d \in D$ is called an upper bound of S if $s \sqsubseteq d$ for every $s \in S$ (notation: $S \sqsubseteq d$).
3. An upper bound d of S is called least upper bound (LUB) or supremum of S if $d \sqsubseteq d'$ for every upper bound d' of S (notation: $d = \bigsqcup S$).

Example 5.7

1. Every subset $S \subseteq \mathbb{N}$ is a chain in $(\mathbb{N}, \leq)$.
It has a LUB (its greatest element) iff it is finite.
2. $\{\emptyset, \{0\}, \{0,1\}, \ldots\}$ is a chain in $(2^{\mathbb{N}}, \subseteq)$ with LUB $\mathbb{N}$.

Chains and Least Upper Bounds

Definition 5.6 (Chain, (least) upper bound)

Let $(D, \sqsubseteq)$ be a partial order and $S \subseteq D$.

1. S is called a chain in D if, for every $s_1, s_2 \in S$,
$$s_1 \sqsubseteq s_2 \text{ or } s_2 \sqsubseteq s_1$$
(that is, S is a totally ordered subset of D).
2. An element $d \in D$ is called an upper bound of S if $s \sqsubseteq d$ for every $s \in S$ (notation: $S \sqsubseteq d$).
3. An upper bound d of S is called least upper bound (LUB) or supremum of S if $d \sqsubseteq d'$ for every upper bound d' of S (notation: $d = \bigsqcup S$).

Example 5.7

1. Every subset $S \subseteq \mathbb{N}$ is a chain in $(\mathbb{N}, \leq)$.
It has a LUB (its greatest element) iff it is finite.
2. $\{\emptyset, \{0\}, \{0, 1\}, \ldots\}$ is a chain in $(2^{\mathbb{N}}, \subseteq)$ with LUB $\mathbb{N}$.

Chain Completeness

Definition 5.8 (Chain completeness)

A partial order is called chain complete (CCPO) if every of its chains has a least upper bound.

Example 5.9

1. $(2^{\mathbb{N}}, \subseteq)$ is a CCPO with $\bigsqcup S = \bigcup_{M \in S} M$ for every chain $S \subseteq 2^{\mathbb{N}}$.
2. $(\mathbb{N}, \leq)$ is not chain complete
   (since, e.g., the chain $\mathbb{N}$ has no upper bound).

Chain Completeness

Definition 5.8 (Chain completeness)

A partial order is called chain complete (CCPO) if every of its chains has a least upper bound.

Example 5.9

1. $(2^{\mathbb{N}}, \subseteq)$ is a CCPO with $\bigsqcup S = \bigcup_{M \in S} M$ for every chain $S \subseteq 2^{\mathbb{N}}$.
2. $(\mathbb{N}, \leq)$ is not chain complete
(since, e.g., the chain $\mathbb{N}$ has no upper bound).

Chain Completeness

Definition 5.8 (Chain completeness)

A partial order is called chain complete (CCPO) if every of its chains has a least upper bound.

Example 5.9

1. $(2^{\mathbb{N}}, \subseteq)$ is a CCPO with $\bigsqcup S = \bigcup_{M \in S} M$ for every chain $S \subseteq 2^{\mathbb{N}}$.
2. $(\mathbb{N}, \leq)$ is not chain complete
   (since, e.g., the chain $\mathbb{N}$ has no upper bound).

Chain Completeness

Definition 5.8 (Chain completeness)

A partial order is called chain complete (CCPO) if every of its chains has a least upper bound.

Example 5.9

1. $(2^{\mathbb{N}}, \subseteq)$ is a CCPO with $\bigsqcup S = \bigcup_{M \in S} M$ for every chain $S \subseteq 2^{\mathbb{N}}$.
2. $(\mathbb{N}, \leq)$ is not chain complete (since, e.g., the chain $\mathbb{N}$ has no upper bound).

Least Elements in CCPOs

Corollary 5.10

Every CCPO has a least element $\bigsqcup \emptyset$.

Proof.

Let $(D, \sqsubseteq)$ be a CCPO.

- By definition, $\emptyset$ is a chain in D.
- By definition, every $d \in D$ is an upper bound of $\emptyset$.
- Thus $\bigsqcup \emptyset$ exists and is the least element of D.

$\square$

Least Elements in CCPOs

Corollary 5.10

Every CCPO has a least element $\bigsqcup \emptyset$.

Proof.

Let $(D, \sqsubseteq)$ be a CCPO.

- By definition, $\emptyset$ is a chain in D.
- By definition, every $d \in D$ is an upper bound of $\emptyset$.
- Thus $\bigsqcup \emptyset$ exists and is the least element of D.

□

Least Elements in CCPOs

Corollary 5.10

Every CCPO has a least element $\bigsqcup \emptyset$.

Proof.

Let $(D, \sqsubseteq)$ be a CCPO.

- By definition, $\emptyset$ is a chain in D.
- By definition, every $d \in D$ is an upper bound of $\emptyset$.
- Thus $\bigsqcup \emptyset$ exists and is the least element of D.

$\square$

Least Elements in CCPOs

Corollary 5.10

Every CCPO has a least element $\bigsqcup \emptyset$.

Proof.

Let $(D, \sqsubseteq)$ be a CCPO.

- By definition, $\emptyset$ is a chain in D.
- By definition, every $d \in D$ is an upper bound of $\emptyset$.
- Thus $\bigsqcup \emptyset$ exists and is the least element of D.

□

Application to fix(Φ) II

Lemma 5.11

- $(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$ *is a CCPO with least element* $f_\emptyset$ *where* $\mathsf{graph}(f_\emptyset) = \emptyset$.
- *In particular, for every chain* $S \subseteq \Sigma \dashrightarrow \Sigma$,

$$\mathsf{graph}\left(\bigsqcup S\right) = \bigcup_{f \in S} \mathsf{graph}(f).$$

Proof.

on the board □

Application to fix(Φ) II

Lemma 5.11

- $(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$ *is a CCPO with least element* $f_\emptyset$ *where* $\mathsf{graph}(f_\emptyset) = \emptyset$.
- *In particular, for every chain* $S \subseteq \Sigma \dashrightarrow \Sigma$,

$$\mathsf{graph}\left(\bigsqcup S\right) = \bigcup_{f \in S} \mathsf{graph}(f).$$

Proof.

on the board □