

Semantics and Verification of Software

Lecture 7: Denotational Semantics of WHILE III (Continuous Functions and Fixpoint Theorem)

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)



noll@cs.rwth-aachen.de

<http://www-i2.informatik.rwth-aachen.de/i2/svsw11/>

Winter Semester 2011/12

Technologie-Beratung @ SAP:

**Über ein spannendes Unternehmen,
innovative Technologien und
Ihren möglichen Berufseinstieg
als Trainee im Consulting**

**SAP Firmenvortrag
Dienstag, 29. November 2011
17:30 bis 18:30
Hörsaal AH1, Ahornstraße**



28. November 2011

Imagination-Innovation-Intervention - Mensch-Roboter Interaktion in der Techno- wissenschaftskultur

Prof. Dr. Jutta Weber - Universität Paderborn

SuperC, 6. Etage, Generali-Saal
17:15 bis 18:45 Uhr



19. Dezember 2011

Von der Hand in den Kopf - Was Hände, Gehirne und Roboter verbindet

Prof. Dr. Helge Ritter - Universität Bielefeld

SuperC, 6. Etage, Generali-Saal
17:15 bis 18:45 Uhr



16. Januar 2012

Morphogenetic Robotics

Prof. Yaochu Jin - University of Surrey

SuperC, 6. Etage, Ford-Saal
17:15 bis 18:45 Uhr



23. Januar 2012

The robotic Scientist: Distilling Natural Laws from Experimental Data, from cognitive ro- botics to computational biology

Prof. Hod Lipson - Cornell University

SuperC, 6. Etage, Generali-Saal
17:15 bis 18:45 Uhr

- 1 Repetition: Denotational Semantics of WHILE
- 2 Monotonic and Continuous Functions
- 3 The Fixpoint Theorem
- 4 An Example
- 5 Summary: Denotational Semantics

Definition (Denotational semantics of statements)

The (denotational) semantic functional for statements,

$$\mathcal{C}[\![\cdot]\!] : \text{Cmd} \rightarrow (\Sigma \dashrightarrow \Sigma),$$

is given by:

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

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

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

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

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

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

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

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

Example

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

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

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

Goals:

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

Sufficient conditions:

on domain $\Sigma \dashrightarrow \Sigma$: **chain-complete partial order**

on function Φ : **continuity**

Lemma

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

Proof.

on the board



Application to $\text{fix}(\Phi)$ II

Lemma

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

Proof.

on the board



Example (cf. Example 6.5(3))

Let $x \in \text{Var}$, and let $f_i : \Sigma \dashrightarrow \Sigma$ for every $i \in \mathbb{N}$ be given by

$$f_i(\sigma) := \begin{cases} \sigma[x \mapsto \sigma(x) + 1] & \text{if } \sigma(x) \leq i \\ \text{undefined} & \text{otherwise} \end{cases}$$

Then $S := \{f_0, f_1, f_2, \dots\}$ is a chain (Example 6.5(3)) with $\bigsqcup S = f$ where

$$f : \Sigma \rightarrow \Sigma : \sigma \mapsto \sigma[x \mapsto \sigma(x) + 1]$$

- 1 Repetition: Denotational Semantics of WHILE
- 2 Monotonic and Continuous Functions
- 3 The Fixpoint Theorem
- 4 An Example
- 5 Summary: Denotational Semantics

Definition 7.1 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders, and let $F : D \rightarrow D'$. F is called **monotonic** (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \implies F(d_1) \sqsubseteq' F(d_2).$$

Definition 7.1 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders, and let $F : D \rightarrow D'$. F is called **monotonic** (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \implies F(d_1) \sqsubseteq' F(d_2).$$

Interpretation: monotonic functions “preserve information”

Definition 7.1 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders, and let $F : D \rightarrow D'$. F is called **monotonic** (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \implies F(d_1) \sqsubseteq' F(d_2).$$

Interpretation: monotonic functions “preserve information”

Example 7.2

- ① Let $T := \{S \subseteq \mathbb{N} \mid S \text{ finite}\}$. Then $F_1 : T \rightarrow \mathbb{N} : S \mapsto \sum_{n \in S} n$ is monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ and $(\mathbb{N}, \leq)$.

Definition 7.1 (Monotonicity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders, and let $F : D \rightarrow D'$. F is called **monotonic** (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \implies F(d_1) \sqsubseteq' F(d_2).$$

Interpretation: monotonic functions “preserve information”

Example 7.2

- Let $T := \{S \subseteq \mathbb{N} \mid S \text{ finite}\}$. Then $F_1 : T \rightarrow \mathbb{N} : S \mapsto \sum_{n \in S} n$ is monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ and $(\mathbb{N}, \leq)$.
- $F_2 : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}} : S \mapsto \mathbb{N} \setminus S$ is not monotonic w.r.t. $(2^{\mathbb{N}}, \subseteq)$ (since, e.g., $\emptyset \subseteq \mathbb{N}$ but $F_2(\emptyset) = \mathbb{N} \not\subseteq F_2(\mathbb{N}) = \emptyset$).

Lemma 7.3

Let $b \in BExp$, $c \in Cmd$, and $\Phi : (\Sigma \dashrightarrow \Sigma) \rightarrow (\Sigma \dashrightarrow \Sigma)$ with $\Phi(f) := \text{cond}(\mathcal{B}[[b]], f \circ \mathcal{C}[[c]], \text{id}_\Sigma)$. Then Φ is monotonic w.r.t. $(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$.

Lemma 7.3

Let $b \in BExp$, $c \in Cmd$, and $\Phi : (\Sigma \dashrightarrow \Sigma) \rightarrow (\Sigma \dashrightarrow \Sigma)$ with $\Phi(f) := \text{cond}(\mathfrak{B}[[b]], f \circ \mathfrak{C}[[c]], \text{id}_\Sigma)$. Then Φ is monotonic w.r.t. $(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$.

Proof.

on the board □

The following lemma states how chains behave under monotonic functions.

Lemma 7.4

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be CCPOs, $F : D \rightarrow D'$ monotonic, and $S \subseteq D$ a chain in D . Then:

- ① $F(S) := \{F(d) \mid d \in S\}$ is a chain in D' .
- ② $\bigsqcup F(S) \sqsubseteq' F(\bigsqcup S)$.

The following lemma states how chains behave under monotonic functions.

Lemma 7.4

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be CCPOs, $F : D \rightarrow D'$ monotonic, and $S \subseteq D$ a chain in D . Then:

- ① $F(S) := \{F(d) \mid d \in S\}$ is a chain in D' .
- ② $\bigsqcup F(S) \sqsubseteq' F(\bigsqcup S)$.

Proof.

on the board



A function F is continuous if the order of applying F and taking LUBs can be reversed:

Definition 7.5 (Continuity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be CCPOs and $F : D \rightarrow D'$ monotonic. Then F is called **continuous** (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every non-empty chain $S \subseteq D$,

$$F \left(\bigsqcup S \right) = \bigsqcup F(S).$$

A function F is continuous if the order of applying F and taking LUBs can be reversed:

Definition 7.5 (Continuity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be CCPOs and $F : D \rightarrow D'$ monotonic. Then F is called **continuous** (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every non-empty chain $S \subseteq D$,

$$F \left(\bigsqcup S \right) = \bigsqcup F(S).$$

Lemma 7.6

Let $b \in BExp$, $c \in Cmd$, and $\Phi(f) := \text{cond}(\mathfrak{B} \llbracket b \rrbracket, f \circ \mathfrak{C} \llbracket c \rrbracket, \text{id}_\Sigma)$. Then Φ is continuous w.r.t. $(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$.

Continuity

A function F is continuous if the order of applying F and taking LUBs can be reversed:

Definition 7.5 (Continuity)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be CCPOs and $F : D \rightarrow D'$ monotonic. Then F is called **continuous** (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every non-empty chain $S \subseteq D$,

$$F \left(\bigsqcup S \right) = \bigsqcup F(S).$$

Lemma 7.6

Let $b \in BExp$, $c \in Cmd$, and $\Phi(f) := \text{cond}(\mathfrak{B} \llbracket b \rrbracket, f \circ \mathfrak{C} \llbracket c \rrbracket, \text{id}_\Sigma)$. Then Φ is continuous w.r.t. $(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$.

Proof.

omitted □

- 1 Repetition: Denotational Semantics of WHILE
- 2 Monotonic and Continuous Functions
- 3 The Fixpoint Theorem**
- 4 An Example
- 5 Summary: Denotational Semantics

The Fixpoint Theorem

Theorem 7.7 (Fixpoint Theorem by Tarski and Knaster)

Let $(D, \sqsubseteq)$ be a CCPO and $F : D \rightarrow D$ continuous. Then

$$\text{fix}(F) := \bigsqcup \left\{ F^n \left(\bigsqcup \emptyset \right) \mid n \in \mathbb{N} \right\}$$

is the least fixpoint of F where

$$F^0(d) := d \text{ and } F^{n+1}(d) := F(F^n(d)).$$

The Fixpoint Theorem

Theorem 7.7 (Fixpoint Theorem by Tarski and Knaster)

Let $(D, \sqsubseteq)$ be a CCPO and $F : D \rightarrow D$ continuous. Then

$$\text{fix}(F) := \bigsqcup \left\{ F^n \left(\bigsqcup \emptyset \right) \mid n \in \mathbb{N} \right\}$$

is the least fixpoint of F where

$$F^0(d) := d \text{ and } F^{n+1}(d) := F(F^n(d)).$$

Proof.

on the board



Application to $\text{fix}(\Phi)$

Altogether this completes the definition of $\mathcal{C}[\![\cdot]\!]$. In particular, for the `while` statement we obtain:

Corollary 7.8

Let $b \in BExp$, $c \in Cmd$, and $\Phi(f) := \text{cond}(\mathfrak{B}[\![b]\!], f \circ \mathcal{C}[\![c]\!], \text{id}_\Sigma)$. Then

$$\text{graph}(\text{fix}(\Phi)) = \bigcup_{n \in \mathbb{N}} \text{graph}(\Phi^n(f_\emptyset))$$

Application to $\text{fix}(\Phi)$

Altogether this completes the definition of $\mathcal{C}[\![\cdot]\!]$. In particular, for the **while** statement we obtain:

Corollary 7.8

Let $b \in BExp$, $c \in Cmd$, and $\Phi(f) := \text{cond}(\mathfrak{B}[\![b]\!], f \circ \mathcal{C}[\![c]\!], \text{id}_\Sigma)$. Then

$$\text{graph}(\text{fix}(\Phi)) = \bigcup_{n \in \mathbb{N}} \text{graph}(\Phi^n(f_\emptyset))$$

Proof.

Using

- Lemma 7.4
 - $(\Sigma \dashrightarrow \Sigma, \sqsubseteq)$ CCPO with least element $f_\emptyset$
 - LUB = union of graphs
- Lemma 7.6 (Φ continuous)
- Theorem 7.7 (Fixpoint Theorem)



- 1 Repetition: Denotational Semantics of WHILE
- 2 Monotonic and Continuous Functions
- 3 The Fixpoint Theorem
- 4 An Example
- 5 Summary: Denotational Semantics

Example 7.9 (Factorial program)

- Let $c \in \text{Cmd}$ be given by
 $y:=1; \text{ while } \neg(x=1) \text{ do } (y:=y*x; x:=x-1)$

Example 7.9 (Factorial program)

- Let $c \in \text{Cmd}$ be given by

$y:=1; \text{ while } \neg(x=1) \text{ do } (y:=y*x; x:=x-1)$

- For every initial state $\sigma_0 \in \Sigma$, Def. 5.3 yields:

$$\mathcal{C}[\![c]\!](\sigma_0) = \text{fix}(\Phi)(\sigma_1)$$

where $\sigma_1 := \sigma_0[y \mapsto 1]$ and, for every $f : \Sigma \dashrightarrow \Sigma$ and $\sigma \in \Sigma$,

$$\begin{aligned}\Phi(f)(\sigma) &= \text{cond}(\mathcal{B}[\![\neg(x=1)]\!], f \circ \mathcal{C}[\![y:=y*x; x:=x-1]\!], \text{id}_\Sigma)(\sigma) \\ &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ f(\sigma') & \text{otherwise} \end{cases}\end{aligned}$$

with $\sigma' := \sigma[y \mapsto \sigma(y) * \sigma(x), x \mapsto \sigma(x) - 1]$.

Example 7.9 (Factorial program)

- Let $c \in \text{Cmd}$ be given by

$y := 1; \text{ while } \neg(x=1) \text{ do } (y := y * x; x := x - 1)$

- For every initial state $\sigma_0 \in \Sigma$, Def. 5.3 yields:

$$\mathcal{C}[[c]](\sigma_0) = \text{fix}(\Phi)(\sigma_1)$$

where $\sigma_1 := \sigma_0[y \mapsto 1]$ and, for every $f : \Sigma \dashrightarrow \Sigma$ and $\sigma \in \Sigma$,

$$\begin{aligned}\Phi(f)(\sigma) &= \text{cond}(\mathcal{B}[[\neg(x=1)]], f \circ \mathcal{C}[[y := y * x; x := x - 1]], \text{id}_\Sigma)(\sigma) \\ &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ f(\sigma') & \text{otherwise} \end{cases}\end{aligned}$$

with $\sigma' := \sigma[y \mapsto \sigma(y) * \sigma(x), x \mapsto \sigma(x) - 1]$.

- Approximations of least fixpoint of Φ according to Theorem 7.7:

$$\text{fix}(\Phi) = \bigsqcup \{ \Phi^n(f_\emptyset) \mid n \in \mathbb{N} \}$$

(where $\text{graph}(f_\emptyset) = \emptyset$)

Example 7.9 (Factorial program; continued)

$$\begin{aligned} f_0(\sigma) &:= \Phi^0(f_\emptyset)(\sigma) \\ &= f_\emptyset(\sigma) \\ &= \text{undefined} \end{aligned}$$

Example 7.9 (Factorial program; continued)

$$\begin{aligned}f_0(\sigma) &:= \Phi^0(f_\emptyset)(\sigma) \\&= f_\emptyset(\sigma) \\&= \text{undefined}\end{aligned}$$

$$\begin{aligned}f_1(\sigma) &:= \Phi^1(f_\emptyset)(\sigma) \\&= \Phi(f_0)(\sigma) \\&= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ f_0(\sigma') & \text{otherwise} \end{cases} \\&= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ \text{undefined} & \text{otherwise} \end{cases}\end{aligned}$$

Example 7.9 (Factorial program; continued)

$$\begin{aligned}
 f_0(\sigma) &:= \Phi^0(f_\emptyset)(\sigma) \\
 &= f_\emptyset(\sigma) \\
 &= \text{undefined} \\
 f_1(\sigma) &:= \Phi^1(f_\emptyset)(\sigma) \\
 &= \Phi(f_0)(\sigma) \\
 &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ f_0(\sigma') & \text{otherwise} \end{cases} \\
 &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ \text{undefined} & \text{otherwise} \end{cases} \\
 f_2(\sigma) &:= \Phi^2(f_\emptyset)(\sigma) \\
 &= \Phi(f_1)(\sigma) \\
 &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ f_1(\sigma') & \text{otherwise} \end{cases} \\
 &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ \sigma' & \text{if } \sigma(x) \neq 1 \text{ and } \sigma'(x) = 1 \\ \text{undefined} & \text{if } \sigma(x) \neq 1 \text{ and } \sigma'(x) \neq 1 \end{cases} \\
 &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ \sigma' & \text{if } \sigma(x) = 2 \\ \text{undefined} & \text{if } \sigma(x) \neq 1 \text{ and } \sigma(x) \neq 2 \end{cases} \\
 &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ \sigma[y \mapsto 2 * \sigma(y), & \text{if } \sigma(x) = 2 \\ \quad x \mapsto 1] & \\ \text{undefined} & \text{if } \sigma(x) \neq 1 \\ & \text{and } \sigma(x) \neq 2 \end{cases}
 \end{aligned}$$

Example 7.9 (Factorial program; continued)

$$\begin{aligned} f_3(\sigma) &:= \Phi^3(f_\emptyset)(\sigma) \\ &= \Phi(f_2)(\sigma) \\ &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ f_2(\sigma') & \text{otherwise} \end{cases} \\ &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ \sigma' & \text{if } \sigma(x) \neq 1 \text{ and } \sigma'(x) = 1 \\ \sigma'[y \mapsto 2 * \sigma'(y), x \mapsto 1] & \text{if } \sigma(x) \neq 1 \text{ and } \sigma'(x) = 2 \\ \text{undefined} & \text{if } \sigma(x) \neq 1 \text{ and } \sigma'(x) \neq 1 \text{ and } \sigma'(x) \neq 2 \end{cases} \\ &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ \sigma' & \text{if } \sigma(x) = 2 \\ \sigma'[y \mapsto 2 * \sigma'(y), x \mapsto 1] & \text{if } \sigma(x) = 3 \\ \text{undefined} & \text{if } \sigma(x) \notin \{1, 2, 3\} \end{cases} \\ &= \begin{cases} \sigma & \text{if } \sigma(x) = 1 \\ \sigma[y \mapsto 2 * \sigma(y), x \mapsto 1] & \text{if } \sigma(x) = 2 \\ \sigma[y \mapsto 3 * 2 * \sigma(y), x \mapsto 1] & \text{if } \sigma(x) = 3 \\ \text{undefined} & \text{if } \sigma(x) \notin \{1, 2, 3\} \end{cases} \end{aligned}$$

Example 7.9 (Factorial program; continued)

- n -th approximation:

$$\begin{aligned} f_n(\sigma) &:= \Phi^n(f_\emptyset)(\sigma) \\ &= \begin{cases} \sigma[y \mapsto \sigma(x) * (\sigma(x) - 1) * \dots * 2 * \sigma(y), \\ \quad x \mapsto 1] & \text{if } 1 \leq \sigma(x) \leq n \\ \text{undefined} & \text{if } \sigma(x) \notin \{1, \dots, n\} \end{cases} \\ &= \begin{cases} \sigma[y \mapsto (\sigma(x))! * \sigma(y), x \mapsto 1] & \text{if } 1 \leq \sigma(x) \leq n \\ \text{undefined} & \text{if } \sigma(x) \notin \{1, \dots, n\} \end{cases} \end{aligned}$$

Example 7.9 (Factorial program; continued)

- n -th approximation:

$$\begin{aligned} f_n(\sigma) &:= \Phi^n(f_\emptyset)(\sigma) \\ &= \begin{cases} \sigma[y \mapsto \sigma(x) * (\sigma(x) - 1) * \dots * 2 * \sigma(y), \\ \quad x \mapsto 1] & \text{if } 1 \leq \sigma(x) \leq n \\ \text{undefined} & \text{if } \sigma(x) \notin \{1, \dots, n\} \end{cases} \\ &= \begin{cases} \sigma[y \mapsto (\sigma(x))! * \sigma(y), x \mapsto 1] & \text{if } 1 \leq \sigma(x) \leq n \\ \text{undefined} & \text{if } \sigma(x) \notin \{1, \dots, n\} \end{cases} \end{aligned}$$

- Fixpoint:

$$\mathcal{C}[\![c]\!](\sigma_0) = \text{fix}(\Phi)(\sigma_1) = \begin{cases} \sigma[y \mapsto (\sigma(x))!, x \mapsto 1] & \text{if } \sigma(x) \geq 1 \\ \text{undefined} & \text{otherwise} \end{cases}$$

- 1 Repetition: Denotational Semantics of WHILE
- 2 Monotonic and Continuous Functions
- 3 The Fixpoint Theorem
- 4 An Example
- 5 Summary: Denotational Semantics**

- Semantic model: partial state transformations ($\Sigma \dashrightarrow \Sigma$)

Summary: Denotational Semantics

- Semantic model: **partial state transformations** ($\Sigma \dashrightarrow \Sigma$)
- **Compositional definition** of functional $\mathcal{C}[\![\cdot]\!] : Cmd \rightarrow (\Sigma \dashrightarrow \Sigma)$

Summary: Denotational Semantics

- Semantic model: **partial state transformations** ($\Sigma \dashrightarrow \Sigma$)
- **Compositional definition** of functional $\mathcal{C}[\![\cdot]\!] : Cmd \rightarrow (\Sigma \dashrightarrow \Sigma)$
- Capturing the recursive nature of loops by a **fixpoint definition** (for a continuous function on a CCPO)

Summary: Denotational Semantics

- Semantic model: **partial state transformations** ($\Sigma \dashrightarrow \Sigma$)
- **Compositional definition** of functional $\mathcal{C}[\![\cdot]\!] : Cmd \rightarrow (\Sigma \dashrightarrow \Sigma)$
- Capturing the recursive nature of loops by a **fixpoint definition** (for a continuous function on a CCPO)
- Approximation by **fixpoint iteration**