Lecture 4: Denotational Semantics – Direct Style Semantics

Concurrency Theory Summer 2024

Dr. Stephan Mennicke

April 23rd, 2024

TU Dresden, Knowledge-Based Systems Group

Review WHILE-Programs

Overview

Part 0: Completing the Introduction

- learning about *bisimilarity* and *bisimulations*

Part 1: Semantics of (Sequential) Programming Languages

- WHILE – an old friend
- denotational semantics (a baseline and an exercise of the inductive method) (**today**)
- natural semantics and (structural) operational semantics

Part 2: Towards Parallel Programming Languages

- bisimilarity and its success story
- deep-dive into induction and coinduction
- algebraic properties of bisimilarity

Part 3: Expressive Power

- Calculus of Communicating Systems (CCS)
- Petri nets

Syntactic Categories

The following categories are pairwise disjoint sets.

- **Num** is the set of numerals (e.g., $n, n_1, n_2, \ldots, \mathtt{0}, \mathtt{1}, \ldots, \mathtt{42}, \ldots$)
- **Var** is the set of variables (e.g., $x, y, z, \ldots$)
- **Aexp** is the set of arithmetic expressions (e.g., $a, a_1 \star a_2, \ldots$)
- **Bexp** is the set of Boolean expressions (e.g., $\mathtt{true}, \neg b, a_1 < a_2, \ldots$)
- **Stm** is the set of all statements (to be defined next)

WHILE Syntax

$$a \;::=\; n \mid x \mid a \oplus a \mid a \star a \mid a \ominus a$$

$$b \;::=\; \texttt{true} \mid \texttt{false} \mid a \equiv a \mid a \leqq a \mid \neg b \mid b \wedge b$$

$$S \;::=\; x \texttt{ := } a \mid \texttt{skip} \mid S \texttt{ ; } S \mid \texttt{if } b \texttt{ then } S \texttt{ else } S \mid \texttt{while } b \texttt{ do } S$$

where $n \in \mathbf{Num}$ and $x \in \mathbf{Var}$.

These are *all* the syntactic categories, rigorously defined by grammars. Really all?

Exercise: Provide a definition for numerals and variables.

Semantic Functions

Assumptions:

1. numerals are given in decimal notation
2. semantic function $\mathcal{N}[\![\cdot]\!] : \mathbf{Num} \to \mathbb{Z}$

In contrast to $\mathbf{Num} = \{\mathtt{0}, \mathtt{1}, -\mathtt{1}, \mathtt{2}, \ldots\}$ we have $\mathbb{Z} = \{0, 1, -1, 2, \ldots\}$

A *state* is a function from variables to $\mathbb{Z}$.

$$\mathbf{State} = \mathbb{Z}^{\mathrm{Var}}$$

Need semantic functions for the syntactic categories

- $\mathcal{A} : \mathbf{Aexp} \to (\,\mathbf{State} \to \mathbb{Z})$
- $\mathcal{B} : \mathbf{Bexp} \to (\,\mathbf{State} \to \mathbb{B})$ (where $\mathbb{B} = \{\mathtt{tt}, \mathtt{ff}\}$)
- $\mathcal{S} : \mathbf{Stm} \to (\,\mathbf{State} \hookrightarrow \mathbf{State})$

Warm-up: Semantics of Expressions

Expressions in a Single Slide

$$
\begin{aligned}
\mathcal{A}[\![n]\!]\,s &:= \mathcal{N}[\![n]\!] \\
\mathcal{A}[\![x]\!]\,s &:= s\,x \\
\mathcal{A}[\![a_1 \oplus a_2]\!]\,s &:= \mathcal{A}[\![a_1]\!]\,s + \mathcal{A}[\![a\_2]\!]\,s \\
\mathcal{A}[\![a_1 \star a_2]\!]\,s &:= \mathcal{A}[\![a_1]\!]\,s \cdot \mathcal{A}[\![a\_2]\!]\,s \\
\mathcal{A}[\![a_1 \ominus a_2]\!]\,s &:= \mathcal{A}[\![a\_1]\!]\,s - \mathcal{A}[\![a\_2]\!]\,s
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{B}[\![\texttt{true}]\!]\,s &:= \texttt{tt} \\
\mathcal{B}[\![\texttt{false}]\!]\,s &:= \texttt{ff} \\
\mathcal{B}[\![a_1 \equiv a_2]\!]\,s &:= \begin{cases} \texttt{tt} \text{ if } \mathcal{A}[\![a_1]\!]\,s = \mathcal{A}[\![a_2]\!]\,s \\ \texttt{ff} \text{ if } \mathcal{A}[\![a_1]\!]\,s \neq \mathcal{A}[\![a_2]\!]\,s \end{cases} \\
\mathcal{B}[\![a_1 \leqq a_2]\!]\,s &:= \begin{cases} \texttt{tt} \text{ if } \mathcal{A}[\![a_1]\!]\,s \leq \mathcal{A}[\![a_2]\!]\,s \\ \texttt{ff} \text{ if } \mathcal{A}[\![a_1]\!]s > \mathcal{A}[\![a_2]\!]\,s \end{cases} \\
\mathcal{B}[\![\neg b]\!]\,s &:= \begin{cases} \texttt{tt} \text{ if } \mathcal{B}[\![b]\!]\,s = \texttt{ff} \\ \texttt{ff} \text{ if } \mathcal{B}[\![b]\!]\,s = \texttt{tt} \end{cases} \\
\mathcal{B}[\![b_1 \wedge b_2]\!]\,s &:= \begin{cases} \texttt{tt} \text{ if } \mathcal{B}[\![b_i]\!]\,s = \texttt{tt} & \text{for } i \in \{1,2\} \\ \texttt{ff} & \text{else.} \end{cases}
\end{aligned}
$$

Properties of Expressions

Definition 1 (Free Variables): For an expression $a \in \mathbf{Aexp}$, define $\mathrm{FV}(a)$ inductively by

- $\mathrm{FV}(n) := \emptyset$,
- $\mathrm{FV}(x) := \{x\}$, and
- $\mathrm{FV}(a_1 \bowtie a_2) := \mathrm{FV}(a_1) \cup \mathrm{FV}(a_2)$ for $\bowtie \in \{\oplus, \star, \ominus\}$.

Theorem 2: Let $s, s' \in \mathbf{State}$ such that $s\,x = s'\,x$ for all $x \in \mathrm{FV}(a)$. Then $\mathcal{A}[\![a]\!]s = \mathcal{A}[\![a]\!]s'$.

Proof: By *structural induction* on a: Base case 1: $a = n$ for some $n \in \mathbf{Num}$, we get $\mathcal{A}[\![a]\!]s = \mathcal{N}[\![n]\!] = \mathcal{A}[\![a]\!]s'$. Base case 2: $a = x$ for some $x \in \mathbf{Var}$, we have (a) $x \in \mathrm{FV}(a)$ (i.e., $s\,x = s'\,x$ by assumption). Hence, $\mathcal{A}[\![a]\!]s = s\,x = s'\,x = \mathcal{A}[\![a]\!]s'$.

For $a = a_1 \bowtie a_2$, we get $\mathrm{FV}(a) = \mathrm{FV}(a_1) \cup \mathrm{FV}(a_2)$ and, by induction hypothesis, $\mathcal{A}[\![a_i]\!]s = \mathcal{A}[\![a_i]\!]s'$ ($i = 1, 2$). Thus,

$$
\begin{aligned}
\mathcal{A}[\![a]\!]s &\overset{\text{(Def.)}}{=} \mathcal{A}[\![a_1 \bowtie a_2]\!]s \\
&\overset{\text{(Def.)}}{=} \mathcal{A}[\![a_1]\!]s \bullet \mathcal{A}[\![a_2]\!]s \\
&\overset{\text{(IH)}}{=} \mathcal{A}[\![a_1]\!]s' \bullet \mathcal{A}[\![a_2]\!]s' \\
&\overset{\text{(Def.)}}{=} \mathcal{A}[\![a_1 \bowtie a_2]\!]s'
\end{aligned}
$$

∎

Semantics of Statements

Semantics of Statements

$$S \;::=\; x \texttt{ := } a \mid \texttt{skip} \mid S \texttt{ ; } S \mid \texttt{if } b \texttt{ then } S \texttt{ else } S \mid \texttt{while } b \texttt{ do } S$$

- aim for function $\mathcal{S}_{\mathsf{ds}} : \mathbf{Stm} \to (\mathbf{State} \hookrightarrow \mathbf{State})$
- $\mathcal{S}_{\mathsf{ds}}[\![x \texttt{ := } a]\!]\, s := s[x \mapsto \mathcal{A}[\![a]\!]\, s]$
- $\mathcal{S}_{\mathsf{ds}}[\![\texttt{skip}]\!] := \mathrm{id}$
- $\mathcal{S}_{\mathsf{ds}}[\![S_1 \texttt{ ; } S_2]\!] := \mathcal{S}_{\mathsf{ds}}[\![S_2]\!] \circ \mathcal{S}_{\mathsf{ds}}[\![S_1]\!]$

$\mathbf{State} \hookrightarrow \mathbf{State}$ is for *partial functions*. For $g : \mathbf{State} \hookrightarrow \mathbf{State}$, we denote that g is *undefined* for value $x \in \mathbf{State}$ by $g\,x = \mathtt{undef}$.

Let $s \in \mathbf{State}$. Then

$$\mathcal{S}_{\mathsf{ds}}[\![S_1\ \mathtt{;}\ S_2]\!]\,s = \begin{cases} s'' & \text{if } s' \text{ exists such that } \mathcal{S}_{\mathsf{ds}}[\![S_1]\!]\,s = s' \text{ and } \mathcal{S}_{\mathsf{ds}}[\![S_2]\!]\,s' = s'' \\ \mathtt{undef} & \text{if } \mathcal{S}_{\mathsf{ds}}[\![S_1]\!]\,s = \mathtt{undef} \text{ or} \\ & \text{if } s' \text{ exists such that } \mathcal{S}_{\mathsf{ds}}[\![S_1]\!]\,s = s' \text{ but } \mathcal{S}_{\mathsf{ds}}[\![S_2]\!]\,s' = \mathtt{undef} \end{cases}$$

Semantics of Conditionals

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

$$\mathsf{cond} : (\mathbf{State} \to \mathbb{B}) \times (\mathbf{State} \hookrightarrow \mathbf{State}) \times (\mathbf{State} \hookrightarrow \mathbf{State}) \to (\mathbf{State} \hookrightarrow \mathbf{State})$$

$$\mathsf{cond}(p, g_1, g_2)s := \begin{cases} g_1\, s \text{ if } p\, s = \texttt{tt} \\ g_2\, s \text{ if } p\, s = \texttt{ff} \end{cases}$$

$$\mathcal{S}_{\mathsf{ds}}[\![\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2]\!]\, s = \begin{cases} s' & \text{if } \mathcal{B}[\![b]\!]\, s = \texttt{tt} \text{ and } s' \text{ exists with } \mathcal{S}_{\mathsf{ds}}[\![S_1]\!]\, s = s' \\ & \text{or if } \mathcal{B}[\![b]\!]\, s = \texttt{ff} \text{ and } s' \text{ exists with } \mathcal{S}_{\mathsf{ds}}[\![S_2]\!]\, s = s' \\ \texttt{undef} & \text{if } \mathcal{B}[\![b]\!] = \texttt{tt} \text{ and } \mathcal{S}_{\mathsf{ds}}[\![S_1]\!]\, s = \texttt{undef} \\ & \text{or if } \mathcal{B}[\![b]\!] = \texttt{ff} \text{ and } \mathcal{S}_{\mathsf{ds}}[\![S_2]\!]\, s = \texttt{undef} \end{cases}$$

Intuition

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

Consequence

Thus, $\mathcal{S}_{\mathsf{ds}}[\![\texttt{while}\ b\ \texttt{do}\ S]\!]$ is a *fixed point* of the functional F:

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

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

We define FIX formally throughout this lecture, but let's first live with our intuition.

Fixed Points by Example

$$\texttt{while}\ \neg(x \equiv 0)\ \texttt{do skip}$$

The corresponding functional is F' such that

$$(F'\,g)s = \begin{cases} g\,s & \text{if } s\,x \neq 0 \\ s & \text{if } s\,x = 0 \end{cases}$$

Surely, g_1 with

$$g_1\,s = \begin{cases} \texttt{undef} & \text{if } s\,x \neq 0 \\ s & \text{if } s\,x = 0 \end{cases}$$

is a fixed point of F' since

Fixed Points by Example

$$
\begin{aligned}
(F'\ g_1)s &= \begin{cases} g_1\ s & \text{if}\ s\,x \neq 0 \\ s & \text{if}\ s\,x = 0 \end{cases} \\
&= \begin{cases} \texttt{undef} & \text{if}\ s\,x \neq 0 \\ s & \text{if}\ s\,x = 0 \end{cases} \\
&= g_1\ s
\end{aligned}
$$

Non-Fixed Points by Example

$$\texttt{while}\ \neg(x \equiv 0)\ \texttt{do skip}$$

The corresponding functional is F' such that

$$(F'\ g)s = \begin{cases} g\ s & \text{if } s\ x \neq 0 \\ s & \text{if } s\ x = 0 \end{cases}$$

Function g_2 such that $g_2\ s = \mathtt{undef}$ for all $s \in \mathbf{State}$ is not a fixed point of F':

For state s' with $s'\ x = 0$, we get $(F'\ g_2)s' = s'$ but $g_2\ s' = \mathtt{undef}$.

- $\mathcal{S}_{\mathsf{ds}}[\![x \texttt{ := } a]\!]\, s := s[x \mapsto \mathcal{A}[\![a]\!]\, s]$
- $\mathcal{S}_{\mathsf{ds}}[\![\texttt{skip}]\!] := \mathrm{id}$
- $\mathcal{S}_{\mathsf{ds}}[\![S_1 \texttt{ ; } S_2]\!] := \mathcal{S}_{\mathsf{ds}}[\![S_2]\!] \circ \mathcal{S}_{\mathsf{ds}}[\![S_1]\!]$
- $\mathcal{S}_{\mathsf{ds}}[\![\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2]\!] := \mathsf{cond}(\mathcal{B}[\![b]\!], S_1, S_2)$
- $\mathcal{S}_{\mathsf{ds}}[\![\texttt{while } b \texttt{ do } S]\!] = \mathsf{FIX}\ F$

Issues to Overcome

1. there are functionals with more than one fixed point (e.g., F')
2. functionals with no fixed point

$$F_1\, g = \begin{cases} g_1 & \text{if } g = g_2 \\ g_2 & \text{otherwise} \end{cases}$$

Requirements on Fixed Points

Consider a statement

$$\texttt{while}\ b\ \texttt{do}\ S$$

from state s_0.

Option A: Termination

Option B: Local Looping

Option C: Global Looping

Option A: Termination

$$\texttt{while}\ b\ \texttt{do}\ S \text{ in state } s_0$$

Then there are states $s_1, \ldots, s_n$ such that

$$\mathcal{B}[\![b]\!]\, s_i = \begin{cases} \texttt{tt} \text{ if } i < n \\ \texttt{ff} \text{ if } i = n \end{cases}$$

and

$$\mathcal{S}_{\mathsf{ds}}[\![S]\!]\, s_i = s_{i+1} \text{ for } i < n$$

Option A: Termination

$$\texttt{while } 0 \leqq \texttt{x do x := x} \ominus 1$$

Let g_0 be any fixed point of F (i.e., $F\,g_0 = g_0$). For $i < n$,

$$\begin{aligned} g_0\,s_i &= (F\,g_0)s_i \\ &= \mathsf{cond}(\,\mathcal{B}[\![0 \leqq \mathsf{x}]\!]\,, g_0 \circ \mathcal{S}_{\mathsf{ds}}[\![\mathsf{x} := \mathsf{x} \ominus 1]\!]\,, \mathrm{id}\,)s_i \\ &= (g_0 \circ \mathcal{S}_{\mathsf{ds}}[\![\mathsf{x} := \mathsf{x} \ominus 1]\!])s_i \\ &= g_0\,s_{i+1} \end{aligned}$$

and for $i = n$,

$$\begin{aligned} g_0\,s_n &= (F\,g_0)s_n \\ &= \mathsf{cond}(\,\mathcal{B}[\![0 \leqq \mathsf{x}]\!]\,, g_0 \circ \mathcal{S}_{\mathsf{ds}}[\![\mathsf{x} := \mathsf{x} \ominus 1]\!]\,, \mathrm{id}\,)s_n \\ &= \mathrm{id}\,s_n = s_n \end{aligned}$$

Every fixed point g of F will satisfy $g\,s_0 = s_n$.

Option B: Local Looping

$$\texttt{while}\ b\ \texttt{do}\ S \text{ in state } s_0$$

Similar observation as before, every fixed point g of F yields $g\,s_0 = \texttt{undef}$.

Exercise: Why?

Option C: Global Looping

$$\texttt{while}\ b\ \texttt{do}\ S \text{ in state } s_0$$

Then there are infinitely many states $s_1, s_2, \ldots$ such that for all $i \geq 0$,

$$\mathcal{B}[\![\neg b]\!]\, s_i = \texttt{tt}$$

and

$$\mathcal{S}_{\mathsf{ds}}[\![S]\!]\, s_i = s_{i+1}$$

Option C: Global Looping

$$\texttt{while}\ \neg(x \equiv \texttt{0})\ \texttt{do}\ \texttt{skip}$$

Let g_0 be any fixed point of F.

We get $g_0\, s_i = g_0\, s_{i+1}$ and, thus,

$$g_0\, s_0 = g_0\, s_i \text{ for all } i \geq 0$$

The functional

$$(F'\, g)s = \begin{cases} g\, s & \text{if } s\, x \neq 0 \\ s & \text{if } s\, x = 0 \end{cases}$$

has various fixed points: every partial function g satisfying $g\, s = s$ if $s\, x = 0$ is one.

Requirements on Fixed Points

Consider a statement

$$\texttt{while}\ b\ \texttt{do}\ S$$

from state s_0.

Option A: Termination

Option B: Local Looping

Option C: Global Looping

Which fixed point to prefer?

Least Fixed Points (if they exist)

Fixed Point Theory

For any function F, we want $\mathsf{FIX}\ F$ to share its result with all other fixed points of F.

Define $\sqsubseteq$ on partial functions **State** $\hookrightarrow$ **State**:

$$g_1 \sqsubseteq g_2 \text{ if } g_1\ s = s' \text{ implies } g_2\ s = s' \text{ for all } s, s' : \textbf{State} \hookrightarrow \textbf{State}.$$

Examples

$$g_1\ s = s \text{ for all } s$$

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

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

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

A *po-set* is a pair $\langle D, \preccurlyeq_D \rangle$ where D is a set and $\preccurlyeq_D$ is a reflexive, transitive, and anti-symmetric binary relation on D.

Lemma 3: If a po-set $\langle D, \preccurlyeq_D \rangle$ has a least element $d \in D$, then d is unique.

Proof: Follows from anti-symmetry of $\preccurlyeq_D$. ∎

The *least element of a poset $\langle D, \preccurlyeq_D \rangle$* is denoted by $\bot_D$ or just $\bot$.

Generally, if $\preccurlyeq_D$ is clear from the context and we just write $\langle D, \preccurlyeq \rangle$

Lemma 4: $\langle \textbf{State} \hookrightarrow \textbf{State}, \sqsubseteq \rangle$ forms a po-set with $\bot$: **State** $\hookrightarrow$ **State**, such that $\bot\ s :=$ `undef` for all s, is its least element.

Now,

1. $\mathsf{FIX}\ F$ is a fixed point of F (i.e., $F(\mathsf{FIX}\ F) = \mathsf{FIX}\ F$), and
2. $\mathsf{FIX}\ F$ is a *least fixed point* of F, meaning $F\,g = g$ implies $\mathsf{FIX}\ F \sqsubseteq g$

But which functionals admit least fixed points?

Completeness for po-sets

For po-set $\langle D, \preccurlyeq \rangle$ and $Y \subseteq D$, we are looking for an element $d \in D$ summarizing all the information in Y.

Such an element d is called *upper bound of Y* if

$$\forall d' \in Y : d' \preccurlyeq d$$

An upper bound d of Y is a *least upper bound* if

for any upper bound d' of Y, we have $d \preccurlyeq d'$.

Lemma 5: If Y has a least upper bound, then it is unique.

Proof: Let $d_1, d_2 \in D$ be least upper bounds of Y, meaning they are upper bounds of Y (i.e., $d \preccurlyeq d_i$ for all $d \in Y$) and they are least under all upper bounds. Hence, $d_1 \preccurlyeq d_2$ and $d_2 \preccurlyeq d_1$. By antisymmetry of $\preccurlyeq$, we get $d_1 = d_2$. ∎

We denote the least upper bound of Y by $\bigsqcup Y$.

(chain-)complete po-sets – (c)cpo

For po-set $\langle D, \preccurlyeq \rangle$ we call $Y \subseteq D$ a *chain* if

$$\text{for any two elements } d_1, d_2 \in Y,\ d_1 \preccurlyeq d_2 \text{ or } d_2 \preccurlyeq d_1.$$

Definition 6: A po-set $\langle D, \preccurlyeq \rangle$ is *chain-complete* (i.e., a chain-complete partially ordered set, or *ccpo*) of $\bigsqcup Y$ exists for all chains $Y \subseteq D$. It is called a *complete lattice* if $\bigsqcup Y$ exists for all subsets Y of D.

Lemma 7: If $\langle D, \preccurlyeq \rangle$ is a ccpo, then it has a least element $\bot$ given by $\bot = \bigsqcup \emptyset$.

Proof: Since $\emptyset$ is (trivially) a chain, $\bigsqcup \emptyset \in D$ by the ccpo property. We need to show that $\bigsqcup \emptyset \preccurlyeq d$ for all $d \in D$.

$$\forall d \in \emptyset : d \preccurlyeq \bigsqcup \emptyset$$

Suppose there was a least element $d_0 \in D - \{\bigsqcup \emptyset\}$. Then $d_0 \preccurlyeq \bigsqcup \emptyset$ and d_0 is an upper bound of $\emptyset$ as well. Since $\bigsqcup \emptyset$ is the least upper bound of $\emptyset$, we get $\bigsqcup \emptyset \preccurlyeq d_0$, entailing $d_0 = \bigsqcup \emptyset$. Hence, $\bigsqcup \emptyset$ is the unique least element $\bot$ of D. ∎

Exercise: Show that **State** $\hookrightarrow$ **State** is not a complete lattice.

An Example Chain

Let g_n : **State** $\hookrightarrow$ **State** be the following partial function

$$g_n\, s = \begin{cases} \texttt{undef} & \text{if } s\, x > n \\ s[x \mapsto -1] & \text{if } 0 \leq s\, x \text{ and } s\, x \leq n \\ s & \text{if } s\, x < 0 \end{cases}$$

It holds that $g_n \preccurlyeq g_m$ whenever $n \leq m$.

The set $Y_0 = \{g_n \mid n \geq 0\}$ is a chain and

$$g\, s = \begin{cases} s[x \mapsto -1] & \text{if } 0 \leq s\, x \\ s & \text{if } s\, x < 0 \end{cases}$$

is its least upper bound $\bigsqcup Y_0$.

Lemma 8: $(\textbf{State} \hookrightarrow \textbf{State}, \sqsubseteq)$ is a ccpo. The least upper bound $\bigsqcup Y$ of a chain Y is given by

$$(\bigsqcup Y)s = s' \text{ if and only if } g\, s = s' \text{ for some } g \in Y.$$

Proof: By Lemma 4, $(\textbf{State} \to \textbf{State}, \sqsubseteq)$ is a po. Let Y be a chain of $\textbf{State} \to \textbf{State}$.

We first show that $\bigsqcup Y$ as defined above is a partial function. Assume, for partial functions $g_1, g_2 \in Y$ we have $g_1\, s = s_1$ and $g_2\, s = s_2$. As Y is a chain, (a) $g_1 \sqsubseteq g_2$ or (b) $g_2 \sqsubseteq g_1$. In either case, we get that $s_1 = s_2$. Thus, $\bigsqcup Y$ is a partial function.

It remains to be shown that $\bigsqcup Y$ is the least upper bound. For function $g \in Y$ and $s \in$ **State** with $g\, s = s'$ we get $(\bigsqcup Y)s = s'$ (by definition). Thus, $\bigsqcup Y$ is an upper bound of g (and of Y). Let g_0 be an upper bound of Y and let $(\bigsqcup Y)s = s'$. Then, by definition of

$\bigsqcup Y$, there is a function $g \in Y$ such that $g\,s = s'$. Hence, $g_0\,s = s'$. This argument holds for all states $s \in \textbf{State}$, entailing $\bigsqcup Y \sqsubseteq g$ (for all upper bounds of Y). ∎

Continuous Functions

Let $\langle D, \preccurlyeq \rangle$ and $\langle D', \preccurlyeq' \rangle$ be ccpo's.

We call a function $f : D \to D'$ monotone if $d_1 \preccurlyeq d_2$ implies $f\,d_1 \preccurlyeq' f\,d_2$ for all $d_1, d_2 \in D$.

Lemma 9: Monotone functions are closed under (functional) composition.

Lemma 10: Let $\langle D, \preccurlyeq \rangle$ and $\langle D', \preccurlyeq' \rangle$ be ccpo's, and let $f : D \to D'$ be monotone. If Y is a chain in D, then $\{f\,d \mid d \in Y\}$ is a chain in D'. Moreover,

$$\bigsqcup{}' \{f\,d \mid d \in Y\} \preccurlyeq' f\Big(\bigsqcup Y\Big)$$

Proof: Define $\Upsilon := \{f\, d \mid d \in Y\}$ and let $d_1', d_2' \in \Upsilon$. Then there are $d_1, d_2 \in D$ such that $f\, d_1 = d_1'$ and $f\, d_2 = d_2'$ (by definition of Υ). Since Y is a chain, it holds that (a) $d_1 \preccurlyeq d_2$ or (b) $d_2 \preccurlyeq d_1$. In case (a), since f is monotone, we get that $d_1' = f\, d_1 \preccurlyeq' f\, d_2 = d_2'$. Case (b) is symmetric. Thus, Υ is a chain.

Let $u = \bigsqcup Y$, i.e. for all $d \in Y$, $d \preccurlyeq u$. As f is monotone, $f\, d \preccurlyeq' f\, u$ for all $d \in Y$. Hence, $f\, u$ is an upper bound of Υ. Since $\bigsqcup' \Upsilon$ is the least upper bound, we get $\bigsqcup' \Upsilon \preccurlyeq' f\, u = f(\bigsqcup Y)$. ■

Exercise: Show that $\bigsqcup' \{f\, d \mid d \in Y\} = f(\bigsqcup Y)$ does not hold in general.

Definition 11: Let $\langle D, \preccurlyeq \rangle$ and $\langle D', \preccurlyeq' \rangle$ be ccpo's. A function $f : D \to D'$ is *continuous* if it is monotone and

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

for all non-empty chains Y of D. If $\bot' = f\ \bot$, then f is called *strict*.

Lemma 12: Continuous functions are closed under (functional) composition.

Theorem 13: Let $f : D \to D$ be a continuous function on the ccpo $\langle D, \preccurlyeq \rangle$ with least element $\bot$. Then

$$\mathsf{FIX}\ f = \bigsqcup \{ f^n \bot \mid n \geq 0 \}$$

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

Proof: Since f is continuous, it is (mon) monotone and (lub) $\bigsqcup \{ f\, d \mid d \in Y \} = f(\bigsqcup Y)$ for all non-empty chains Y.

First observe that $\{ f^n \bot \mid n \geq 0 \}$ is non-empty by $f^0 \bot = \bot$. It holds that $f^0 \bot = \bot \preccurlyeq f^1 \bot = f \bot$ since $\bot$ is the least element of D. By an inductive argument, we get that $f^m \bot \preccurlyeq f^{m+1} \bot$ for all $m \geq 0$ since f is monotone. By reflexivity and transitivity of $\preccurlyeq$ we get $f^m \bot \preccurlyeq f^n \bot$ whenever $m \leq n$. Therefore, $\{ f^n \bot \mid n \geq 0 \}$ is a non-empty chain and, thus, $\bigsqcup\{f^n \perp \mid n \geq 0\}$ exists (i.e., defines an element of D). We next show that it is a fixed point of f:

$$\begin{aligned} f\Big(\bigsqcup\{f^n \perp \mid n \geq 0\}\Big) &= \bigsqcup\{f(f^n) \perp \mid n \geq 0\} \\ &= \bigsqcup\{f^n \perp \mid n \geq 1\} \\ &= \bigsqcup(\{f^n \perp \mid n \geq 1\} \cup \{\perp\}) \\ &= \bigsqcup\{f^n \perp \mid n \geq 0\} \end{aligned}$$

It remains to be shown that $\mathsf{FIX}\ f$ is the least fixed point of f. For an arbitrary fixed point d of f, we have that $f\,d = d$ and, clearly, $\perp \preccurlyeq d$. By monotonicity of f and an induction on n, we get $f^n \perp \preccurlyeq f^n\,d = d$ for all $n \geq 0$. Hence, d is an upper bound for the chain $\{f^n \perp \mid n \geq 0\}$ and since $\mathsf{FIX}\ f$ is the least upper bound of that chain, we directly obtain $\mathsf{FIX}\ f \preccurlyeq d$. ■

What Remains to be Shown

What Remains to be Shown

- $\mathcal{S}_{\mathsf{ds}}[\![x \texttt{ := } a]\!]\, s := s[x \mapsto \mathcal{A}[\![a]\!]\, s]$
- $\mathcal{S}_{\mathsf{ds}}[\![\texttt{skip}]\!] := \mathrm{id}$
- $\mathcal{S}_{\mathsf{ds}}[\![S_1 \texttt{ ; } S_2]\!] := \mathcal{S}_{\mathsf{ds}}[\![S_1]\!] \circ \mathcal{S}_{\mathsf{ds}}[\![S_1]\!]$
- $\mathcal{S}_{\mathsf{ds}}[\![\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2]\!] := \mathsf{cond}(\mathcal{B}[\![b]\!], S_1, S_2)$
- $\mathcal{S}_{\mathsf{ds}}[\![\texttt{while } b \texttt{ do } S]\!]\, s = \mathsf{FIX}\ F$

1. Functionals F are continuous
2. The direct style semantics $\mathcal{S}_{\mathsf{ds}}[\![\cdot]\!]$ exists