Korrekte Software: Grundlagen und Methoden

Vorlesung 3 vom 05.05.20

Denotationale Semantik

Serge Autexier, Christoph Lüth

Universität Bremen

Sommersemester 2020

Fahrplan

- Einführung
- Operationale Semantik
- Denotationale Semantik
- Äquivalenz der Operationalen und Denotationalen Semantik
- Der Floyd-Hoare-Kalkül
- Invarianten und die Korrektheit des Floyd-Hoare-Kalküls
- Strukturierte Datentypen
- Verifikationsbedingungen
- Vorwärts mit Floyd und Hoare
- Modellierung
- Spezifikation von Funktionen
- Referenzen und Speichermodelle
- Ausblick und Rückblick

Überblick

- Denotationale Semantik für C0
- Fixpunkte

Denotationale Semantik — Motivation

- **Operationale Semantik:**

  Eine Menge von Regeln, die einen Zustand und ein Programm in einen neuen Zustand oder Fehler überführen

$$\langle c, \sigma \rangle \to_{Stmt} \sigma' | \bot$$

- **Denotationale Semantik:**

  Eine Menge von Regeln, die ein Programm in eine $\underbrace{\textbf{partielle Funktion}}_{\text{Denotat}}$ von Zustand nach Zustand überführen

$$[\![c]\!]_{\mathcal{C}} : \Sigma \rightharpoonup \Sigma$$

Denotationale Semantik — Motivation

Zwei Programme sind äquivalent gdw. sie immer zum selben Zustand (oder Fehler) auswerten

$$c_0 \sim c_1 \text{ iff } (\forall \sigma, \sigma'. \langle c_0, \sigma \rangle \rightarrow_{Stmt} \sigma' \Leftrightarrow \langle c_1, \sigma \rangle \rightarrow_{Stmt} \sigma')$$

oder

Zwei Programme sind äquivalent gdw. sie dieselbe partielle Funktion **denotieren**

$$c_0 \sim c_1 \text{ iff } \{(\sigma, \sigma') | \langle c_0, \sigma \rangle \rightarrow_{Stmt} \sigma'\} = \{(\sigma, \sigma') | \langle c_1, \sigma \rangle \rightarrow_{Stmt} \sigma'\}$$

Kompositionalität

- Semantik von zusammengesetzten Ausdrücken durch Kombination der Semantiken der Teilausdrücke
  - Bsp: Semantik einer Sequenz von Anweisungen durch Verknüpfung der Semantik der einzelnen Anweisungen
- Operationale Semantik ist **nicht** kompositional:

```
x= 3;
y= x+ 7; // (*)
z= x+ y;
```

- Semantik von Zeile (*) ergibt sich aus der Ableitung davor
- Kann nicht unabhängig abgeleitet werden

- Denotationale Semantik ist kompositional.
  - Wesentlicher Baustein: **partielle Funktionen**

Partielle Funktion

Definition (Partielle Funktion)

Eine **partielle Funktion** $f : X \rightharpoonup Y$ ist eine Relation $f \subseteq X \times Y$ so dass wenn $(x, y_1) \in f$ und $(x, y_2) \in f$ dann $y_1 = y_2$ (**Rechtseindeutigkeit**)

- Notation: für $f : X \rightharpoonup Y$, $(x, y) \in f \iff f(x) = y$.
- Wir benutzen beide Notationen, aber für die denotationale Semantik die Paar-Notation.
- Zustände sind partielle Abbildungen ($\longrightarrow$ letzte Vorlesung)
- Insbesondere **Systemzustände** $\Sigma = \mathbf{Loc} \rightharpoonup \mathbf{V}$

Denotierende Funktionen

- Arithmetische Ausdrücke:

  $a \in \mathbf{Aexp}$ denotiert eine partielle Funktion $\Sigma \rightharpoonup \mathbb{Z}$

- Boolsche Ausdrücke:

  $b \in \mathbf{Bexp}$ denotiert eine partielle Funktion $\Sigma \rightharpoonup \mathbb{B}$

- Anweisungen:

  $c \in \mathbf{Stmt}$ denotiert eine partielle Funktion $\Sigma \rightharpoonup \Sigma$

Denotat von Aexp

$$[\![a]\!]_{\mathcal{A}} : \textbf{Aexp} \to (\Sigma \rightharpoonup \mathbb{Z})$$

$$
\begin{aligned}
[\![n]\!]_{\mathcal{A}} &= \{(\sigma, n) \mid \sigma \in \Sigma\} \\
[\![x]\!]_{\mathcal{A}} &= \{(\sigma, \sigma(x)) \mid \sigma \in \Sigma, x \in Dom(\sigma)\} \\
[\![a_0 + a_1]\!]_{\mathcal{A}} &= \{(\sigma, n_0 + n_1) \mid (\sigma, n_0) \in [\![a_0]\!]_{\mathcal{A}} \wedge (\sigma, n_1) \in [\![a_1]\!]_{\mathcal{A}}\} \\
[\![a_0 - a_1]\!]_{\mathcal{A}} &= \{(\sigma, n_0 - n_1) \mid (\sigma, n_0) \in [\![a_0]\!]_{\mathcal{A}} \wedge (\sigma, n_1) \in [\![a_1]\!]_{\mathcal{A}}\} \\
[\![a_0 * a_1]\!]_{\mathcal{A}} &= \{(\sigma, n_0 * n_1) \mid (\sigma, n_0) \in [\![a_0]\!]_{\mathcal{A}} \wedge (\sigma, n_1) \in [\![a_1]\!]_{\mathcal{A}}\} \\
[\![a_0 / a_1]\!]_{\mathcal{A}} &= \{(\sigma, n_0 / n_1) \mid (\sigma, n_0) \in [\![a_0]\!]_{\mathcal{A}} \wedge (\sigma, n_1) \in [\![a_1]\!]_{\mathcal{A}} \wedge n_1 \neq 0\}
\end{aligned}
$$

Rechtseindeutigkeit

Lemma (Partielle Funktion)

*$[\![-]\!]_{\mathcal{A}}$ ist rechtseindeutig und damit eine **partielle Funktion**.*

Beweis:

z.z.: wenn $(\sigma, v_1) \in [\![a]\!]_{\mathcal{A}}, (\sigma, v_2) \in [\![a]\!]_{\mathcal{A}}$ dann $v_1 = v_2$.

Strukturelle Induktion über **Aexp**:

Rechtseindeutigkeit

Lemma (Partielle Funktion)

$[\![-]\!]_{\mathcal{A}}$ *ist rechtseindeutig und damit eine* ***partielle Funktion****.*

Beweis:

z.z.: wenn $(\sigma, v_1) \in [\![a]\!]_{\mathcal{A}}, (\sigma, v_2) \in [\![a]\!]_{\mathcal{A}}$ dann $v_1 = v_2$.
Strukturelle Induktion über **Aexp**:

- Induktionsbasis sind $n \in \mathbf{Z}$ und $x \in \mathbf{Idt}$.
  Sei $a \equiv x$, dann $v_1 = \sigma(x) = v_2$.

Rechtseindeutigkeit

Lemma (Partielle Funktion)

$[\![-]\!]_{\mathcal{A}}$ *ist rechtseindeutig und damit eine **partielle Funktion**.*

Beweis:

z.z.: wenn $(\sigma, v_1) \in [\![a]\!]_{\mathcal{A}}, (\sigma, v_2) \in [\![a]\!]_{\mathcal{A}}$ dann $v_1 = v_2$.
Strukturelle Induktion über **Aexp**:

- Induktionsbasis sind $n \in \mathbf{Z}$ und $x \in \mathbf{Idt}$.
  Sei $a \equiv x$, dann $v_1 = \sigma(x) = v_2$.

- Induktionssschritt sind die anderen Klauseln.
  Sei $a \equiv a_1 + a_2$.
  Induktionsannahme ist $(\sigma, n_i) \in [\![a_i]\!]_{\mathcal{A}}, (\sigma, n_i') \in [\![a_i]\!]_{\mathcal{A}}$ dann $n_i = n_i'$.
  Dann $v_1 = (\sigma, n_1 + n_2)$ mit $(\sigma, n_1) \in [\![a_1]\!]_{\mathcal{A}}, (\sigma, n_2) \in [\![a_2]\!]_{\mathcal{A}}$, und $v_2 = n_1' + n_2'$ mit $(\sigma, n_1') \in [\![a_1]\!]_{\mathcal{A}}, (\sigma, n_2') \in [\![a_2]\!]_{\mathcal{A}}$. Aus der Annahme folgt $n_1 = n_1'$ und $n_2 = n_2'$, deshalb $v_1 = v_2$.

$\square$

Kompositionalität und Striktheit

- Die Rechtseindeutigkeit erlaubt die Notation als partielle Funktion:

$$\begin{aligned}
[\![3 * (x + y)]\!]_{\mathcal{A}}(\sigma) &= [\![3]\!]_{\mathcal{A}}(\sigma) \cdot ([\![x]\!]_{\mathcal{A}}(\sigma) + [\![y]\!]_{\mathcal{A}}(\sigma)) \\
&= 3 \cdot ([\![x]\!]_{\mathcal{A}}(\sigma) + [\![y]\!]_{\mathcal{A}}(\sigma)) \\
&= 3 \cdot (\sigma(x) + \sigma(y))
\end{aligned}$$

- Diese Notation versteckt die **Partialität**:

$$[\![1 + x/0]\!]_{\mathcal{A}}(\sigma) = 1 + \sigma(x)/0 = 1 + \perp = \perp$$

- Wenn ein Teilausdruck undefiniert ist, wird der gesamte Ausdruck undefiniert: $[\![-]\!]_{\mathcal{A}}$ ist **strikt** für alle arithmetischen Operatoren.

Arbeitsblatt 3.1: Semantik I

Hier üben wir noch einmal den Zusammenhang zwischen den beiden Notationen. Gegeben sei der Zustand $s = \langle x \mapsto 3, y \mapsto 4\rangle$ und der Ausdruck $a = 7 * x + y$.

Berechnen Sie die Semantik zum einen als Relation (füllen Sie die Fragezeichen aus):

```
(s, ?) : [[7]]
(s, ?) : [[x]]
(s, ?) : [[7*x]]
(s, ?) : [[y]]
(s, ?) : [[7*x+ y]]
```

Berechnen Sie zum anderen die Semantik in der Funktionsnotation:

```
[[7*x+y]](s) = [[7*x]](s)+[[y]](s) = ... = ?
```

Ist das Ergebnis am Ende gleich?

Denotat von Bexp

$$\llbracket a \rrbracket_{\mathcal{B}} : \textbf{Bexp} \rightarrow (\Sigma \rightharpoonup \mathbb{B})$$

$$\begin{aligned}
\llbracket 1 \rrbracket_{\mathcal{B}} &= \{(\sigma, \mathit{true}) \mid \sigma \in \Sigma\} \\
\llbracket 0 \rrbracket_{\mathcal{B}} &= \{(\sigma, \mathit{false}) \mid \sigma \in \Sigma\} \\
\llbracket a_0 == a_1 \rrbracket_{\mathcal{B}} &= \{(\sigma, \mathit{true}) \mid \sigma \in \Sigma, (\sigma, n_0) \in \llbracket a_0 \rrbracket_{\mathcal{A}}(\sigma), \\
&\qquad (\sigma, n_1) \in \llbracket a_1 \rrbracket_{\mathcal{A}}, n_0 = n_1\} \\
&\quad \cup \{(\sigma, \mathit{false}) \mid \sigma \in \Sigma, (\sigma, n_0) \in \llbracket a_0 \rrbracket_{\mathcal{A}}(\sigma), \\
&\qquad (\sigma, n_1) \in \llbracket a_1 \rrbracket_{\mathcal{A}}, n_0 \neq n_1\} \\
\llbracket a_0 < a_1 \rrbracket_{\mathcal{B}} &= \{(\sigma, \mathit{true}) \mid \sigma \in \Sigma, (\sigma, n_0) \in \llbracket a_0 \rrbracket_{\mathcal{A}}(\sigma), \\
&\qquad (\sigma, n_1) \in \llbracket a_1 \rrbracket_{\mathcal{A}}, n_0 < n_1\} \\
&\quad \cup \{(\sigma, \mathit{false}) \mid \sigma \in \Sigma, (\sigma, n_0) \in \llbracket a_0 \rrbracket_{\mathcal{A}}(\sigma), \\
&\qquad (\sigma, n_1) \in \llbracket a_1 \rrbracket_{\mathcal{A}}, n_0 \geq n_1\}
\end{aligned}$$

Denotat von Bexp

$$[\![a]\!]_{\mathcal{B}} : \textbf{Bexp} \to (\Sigma \rightharpoonup \mathbb{B})$$

$$\begin{aligned}
[\![!b]\!]_{\mathcal{B}} &= \{(\sigma, \mathit{true}) \mid \sigma \in \Sigma, (\sigma, \mathit{false}) \in [\![b]\!]_{\mathcal{B}}\} \\
&\quad \cup \{(\sigma, \mathit{false}) \mid \sigma \in \Sigma, (\sigma, \mathit{true}) \in [\![b]\!]_{\mathcal{B}}\} \\
[\![b_1 \;\&\&\; b2]\!]_{\mathcal{B}} &= \{(\sigma, \mathit{false}) \mid \sigma \in \Sigma, (\sigma, \mathit{false}) \in [\![b_1]\!]_{\mathcal{B}}\} \\
&\quad \cup \{(\sigma, t_2) \mid \sigma \in \Sigma, (\sigma, \mathit{true}) \in [\![b_1]\!]_{\mathcal{B}}, (\sigma, t_2) \in [\![b_2]\!]_{\mathcal{B}}\} \\
[\![b_1 \mid\mid b_2]\!]_{\mathcal{B}} &= \{(\sigma, \mathit{true}) \mid \sigma \in \Sigma, (\sigma, \mathit{true}) \in [\![b_1]\!]_{\mathcal{B}}\} \\
&\quad \cup \{(\sigma, t_2) \mid \sigma \in \Sigma, (\sigma, \mathit{false}) \in [\![b_1]\!]_{\mathcal{B}}, (\sigma, t_2) \in [\![b_2]\!]_{\mathcal{B}}\}
\end{aligned}$$

Kompositionalität und Striktheit

Lemma (Partielle Funktion)

$[\![-]\!]_{\mathcal{B}}$ *ist rechtseindeutig und damit eine* ***partielle Funktion****.*

- Beweis analog zu $[\![-]\!]_{\mathcal{A}}$.
- Ist $[\![-]\!]_{\mathcal{B}}$ strikt?

Kompositionalität und Striktheit

Lemma (Partielle Funktion)

$[\![-]\!]_{\mathcal{B}}$ *ist rechtseindeutig und damit eine* ***partielle Funktion****.*

- Beweis analog zu $[\![-]\!]_{\mathcal{A}}$.
- Ist $[\![-]\!]_{\mathcal{B}}$ strikt? Natürlich nicht:
- Sei $[\![b_1]\!]_{\mathcal{B}}(\sigma) = \mathit{false}$, dann $[\![b_1 \ \&\& \ b_2]\!]_{\mathcal{B}}(\sigma) = [\![b_1]\!]_{\mathcal{B}}(\sigma) = \mathit{false}$

Kompositionalität und Striktheit

Lemma (Partielle Funktion)

$[\![-]\!]_{\mathcal{B}}$ *ist rechtseindeutig und damit eine* ***partielle Funktion****.*

- Beweis analog zu $[\![-]\!]_{\mathcal{A}}$.
- Ist $[\![-]\!]_{\mathcal{B}}$ strikt? Natürlich nicht:
- Sei $[\![b_1]\!]_{\mathcal{B}}(\sigma) = \mathit{false}$, dann $[\![b_1 \;\&\&\; b_2]\!]_{\mathcal{B}}(\sigma) = [\![b_1]\!]_{\mathcal{B}}(\sigma) = \mathit{false}$
- Wir können deshalb nicht so einfach schreiben $[\![b_1 \;\&\&\; b_2]\!]_{\mathcal{B}}(\sigma) = [\![b_1]\!]_{\mathcal{B}}(\sigma) \wedge [\![b_2]\!]_{\mathcal{B}}(\sigma)$
- Die normale zweiwertige Logik behandelt Definiertheit gar nicht. Bei uns müssen die logischen Operatoren links-strikt sein:

$$
\begin{array}{lll}
\bot \wedge a = \bot & \mathit{false} \wedge a = \mathit{false} & \mathit{true} \wedge a = a \\
\bot \vee a = \bot & \mathit{true} \vee a = \mathit{true} & \mathit{false} \vee a = a
\end{array}
$$

Arbeitsblatt 3.2: Semantik II

Wir üben noch einmal die Nichtstrikheit. Gegeben $s = \langle x \mapsto 7 \rangle$ und $b = (7 == x) \,||\, (x/0 == 1)$
Berechnenen Sie die Semantik als Relation in der Notation von oben:

```
(s, ?) : [[ (7 == x) || (x/0 == 1) ]]
...

[[ (7 == x) || (x/0 == 1) ]] = ?
```

Hilfreiche Notation: $a \wedge b =$ `a /\ b`, $a \vee b =$ `a \/ b`

Denotationale Semantik von Anweisungen

- Zuweisung: punktuelle Änderung des Zustands $\sigma \mapsto \sigma[n/x]$
- Sequenz: Komposition von Relationen

Definition (Komposition von Relationen)

Für zwei Relationen $R \subseteq X \times Y, S \subseteq Y \times Z$ ist ihre **Komposition**

$$R \circ S \stackrel{def}{=} \{(x, z) \mid \exists y \in Y. (x, y) \in R \wedge (y, z) \in S\}$$

Wenn R, S zwei partielle Funktionen sind, ist $R \circ S$ ihre Funktionskomposition.

- Leere Sequenz: Leere Funktion?

Denotationale Semantik von Anweisungen

- Zuweisung: punktuelle Änderung des Zustands $\sigma \mapsto \sigma[n/x]$
- Sequenz: Komposition von Relationen

Definition (Komposition von Relationen)

Für zwei Relationen $R \subseteq X \times Y, S \subseteq Y \times Z$ ist ihre **Komposition**

$$R \circ S \stackrel{def}{=} \{(x, z) \mid \exists y \in Y. (x, y) \in R \wedge (y, z) \in S\}$$

Wenn R, S zwei partielle Funktionen sind, ist $R \circ S$ ihre Funktionskomposition.

- Leere Sequenz: Leere Funktion? Nein, Identität. Für Menge X,

$$\mathbf{Id}_X \stackrel{def}{=} X \times X = \{(x, x) \mid x \in X\}q$$

ist die **Identitätsfunktion** ($\mathbf{Id}_X(x) = x$).

Arbeitsblatt 3.3: Komposition von Relationen

Zur Übung: betrachten Sie folgende Relationen:

$$R = \{(1,7),(2,3),(3,9),(4,3)\}$$
$$S = \{(1,0),(2,0),(3,1),(4,7),(5,9),(7,3),(8,15)\}$$

Berechnen Sie $R \circ S = \{(1,?),\ldots\}$

Denotat von Stmt

$$[\![.]\!]_{\mathcal{C}} : \textbf{Stmt} \to (\Sigma \rightharpoonup \Sigma)$$

$$\begin{array}{rcl}
[\![x = a]\!]_{\mathcal{C}} & = & \{(\sigma, \sigma[n/x]) \mid \sigma \in \Sigma \land (\sigma, n) \in [\![a]\!]_{\mathcal{A}}\} \\
[\![c_1; c_2]\!]_{\mathcal{C}} & = & [\![c_1]\!]_{\mathcal{C}} \circ [\![c_2]\!]_{\mathcal{C}} \\
[\![\{\ \}]\!]_{\mathcal{C}} & = & \mathbf{Id}_{\Sigma} \\
[\![\textbf{if}\ (b)\ c_0\ \textbf{else}\ c_1]\!]_{\mathcal{C}} & = & \{(\sigma, \sigma') \mid (\sigma, \mathit{true}) \in [\![b]\!]_{\mathcal{B}} \land (\sigma, \sigma') \in [\![c_0]\!]_{\mathcal{C}}\} \\
& & \cup \{(\sigma, \sigma') \mid (\sigma, \mathit{false}) \in [\![b]\!]_{\mathcal{B}} \land (\sigma, \sigma') \in [\![c_1]\!]_{\mathcal{C}}\}
\end{array}$$

Denotat von Stmt

$$[\![.]\!]_{\mathcal{C}} : \textbf{Stmt} \to (\Sigma \rightharpoonup \Sigma)$$

$$\begin{array}{rcl}
[\![x = a]\!]_{\mathcal{C}} & = & \{(\sigma, \sigma[n/x]) \mid \sigma \in \Sigma \land (\sigma, n) \in [\![a]\!]_{\mathcal{A}}\} \\
[\![c_1; c_2]\!]_{\mathcal{C}} & = & [\![c_1]\!]_{\mathcal{C}} \circ [\![c_2]\!]_{\mathcal{C}} \\
[\![\{\ \}]\!]_{\mathcal{C}} & = & \textbf{Id}_{\Sigma} \\
[\![\textbf{if}\ (b)\ c_0\ \textbf{else}\ c_1]\!]_{\mathcal{C}} & = & \{(\sigma, \sigma') \mid (\sigma, \mathit{true}) \in [\![b]\!]_{\mathcal{B}} \land (\sigma, \sigma') \in [\![c_0]\!]_{\mathcal{C}}\} \\
 & & \cup\{(\sigma, \sigma') \mid (\sigma, \mathit{false}) \in [\![b]\!]_{\mathcal{B}} \land (\sigma, \sigma') \in [\![c_1]\!]_{\mathcal{C}}\}
\end{array}$$

Aber was ist

$$[\![\textbf{while}\ (b)\ c]\!]_{\mathcal{C}} = ??$$

Denotationale Semantik von while

- Sei $w \equiv \textbf{while}\ (b)\ c$ (und $\sigma \in \Sigma$). Operational gilt:

$$w \sim \textbf{if}\ (b)\ \{c; w\}\ \textbf{else}\ \{\,\}$$

- Dann sollte auch gelten

$$[\![w]\!]_{\mathcal{C}} \stackrel{?}{=} [\![\textbf{if}\ (b)\ \{c; w\}\ \textbf{else}\ \{\,\}]\!]_{\mathcal{C}}$$

- Das ist eine **rekursive** Definition von $[\![w]\!]_{\mathcal{C}}$:

$$x = F(x)$$

- Das ist ein **Fixpunkt**:

$$x = \mathit{fix}(F)$$

- Was ist das?

Fixpunkte

Definition (Fixpunkt)

Für $f : X \rightharpoonup X$ ist ein **Fixpunkt** ein $x \in X$ so dass $f(x) = x$.

- Hat jede Funktion $f : X \rightharpoonup X$ einen Fixpunkt?

Fixpunkte

Definition (Fixpunkt)

Für $f : X \rightharpoonup X$ ist ein **Fixpunkt** ein $x \in X$ so dass $f(x) = x$.

- Hat jede Funktion $f : X \rightharpoonup X$ einen Fixpunkt? Nein
- Kann eine Funktion mehrere Fixpunkte haben?

Fixpunkte

Definition (Fixpunkt)

Für $f : X \rightharpoonup X$ ist ein **Fixpunkt** ein $x \in X$ so dass $f(x) = x$.

- Hat jede Funktion $f : X \rightharpoonup X$ einen Fixpunkt? Nein
- Kann eine Funktion mehrere Fixpunkte haben? Ja — aber nur einen kleinsten.
- Beispiele
  - Fixpunkte von $f(x) = \sqrt{x}$ sind 0 und 1; ebenfalls für $f(x) = x^2$.
  - Für die Sortierfunktion sind alle sortierten Listen Fixpunkte
  - Die Funktion $f(x) = x + 1$ hat keinen Fixpunkt in $\mathbb{Z}$
  - Die Funktion $f(X) = \mathbb{P}(X)$ hat überhaupt keinen Fixpunkt
- $\mathit{fix}(f)$ ist also der **kleinste Fixpunkt** von f.

Konstruktion des kleinsten Fixpunktes (Kurzversion)

- Gegeben Funktion Γ auf Denotaten $\Gamma : (\Sigma \rightharpoonup \Sigma) \rightharpoonup (\Sigma \rightharpoonup \Sigma)$
- Wir konstruieren eine Sequenz $\Gamma^i : \Sigma \rightharpoonup \Sigma$ (mit $i \in \mathbb{N}$) von Funktionen:

$$\Gamma^0(s) \stackrel{def}{=} \emptyset$$
$$\Gamma^{i+1}(s) \stackrel{def}{=} \Gamma(\Gamma^i(s))$$

- Dann ist

$$fix(\Gamma) \stackrel{def}{=} \bigcup_{i \in \mathbb{N}} \Gamma^i$$

- Verkürzte Version — der Fixpunkt muss so nicht existieren (er tut es aber für alle Programme)

Denotationale Semantik für die Iteration

- Sei $w \equiv \textbf{while}\ (b)\ c$
- Konstruktion: "Auffalten" der Schleife (s ist ein Denotat):

$$
\begin{aligned}
\Gamma(s) = &\{(\sigma, \sigma') \mid (\sigma, \mathit{true}) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma, \sigma') \in [\![c]\!]_{\mathcal{C}} \circ s\} \\
&\cup \{(\sigma, \sigma) \mid (\sigma, \mathit{false}) \in [\![b]\!]_{\mathcal{B}}\}
\end{aligned}
$$

- b und c sind Parameter von Γ
- Dann ist

$$[\![w]\!]_{\mathcal{C}} = \mathit{fix}(\Gamma)$$

Denotation für Stmt

$$[\![.]\!]_{\mathcal{C}} : \textbf{Stmt} \rightarrow (\Sigma \rightharpoonup \Sigma)$$

$$\begin{aligned}
[\![x = a]\!]_{\mathcal{C}} =& \{(\sigma, \sigma[n/x]) \mid \sigma \in \Sigma \wedge (\sigma, n) \in [\![a]\!]_{\mathcal{A}}\} \\
[\![c_1; c_2]\!]_{\mathcal{C}} =& [\![c_1]\!]_{\mathcal{C}} \circ [\![c_2]\!]_{\mathcal{C}} \\
[\![\{\ \}]\!]_{\mathcal{C}} =& \mathbf{Id}_{\Sigma} \\
[\![\textbf{if}\ (b)\ c_0\ \textbf{else}\ c_1]\!]_{\mathcal{C}} =& \{(\sigma, \sigma') \mid (\sigma, \mathit{true}) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma, \sigma') \in [\![c_0]\!]_{\mathcal{C}}\} \\
& \cup \{(\sigma, \sigma') \mid (\sigma, \mathit{false}) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma, \sigma') \in [\![c_1]\!]_{\mathcal{C}}\} \\
[\![\textbf{while}\ (b)\ c]\!]_{\mathcal{C}} =& \mathit{fix}(\Gamma)
\end{aligned}$$

$$\begin{aligned}
\Gamma(s) =& \{(\sigma, \sigma') \mid (\sigma, \mathit{true}) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma, \sigma') \in [\![c]\!]_{\mathcal{C}} \circ s\} \\
& \cup \{(\sigma, \sigma) \mid (\sigma, \mathit{false}) \in [\![b]\!]_{\mathcal{B}}\}
\end{aligned}$$

Der Fixpunkt bei der Arbeit (I)

```
while (x < 0) {
  x= x+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} \begin{cases} \sigma & \sigma(x) \geq 0 \\ f(\sigma[\sigma(x)+1/x]) & \sigma(x) < 0 \end{cases}$$

Wir betrachten den Zustand $s = \langle x \mapsto ? \rangle$ (nur eine Variable):

Der Fixpunkt bei der Arbeit (I)

```
while (x < 0) {
  x= x+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} \begin{cases} \sigma & \sigma(x) \geq 0 \\ f(\sigma[\sigma(x) + 1/x]) & \sigma(x) < 0 \end{cases}$$

Wir betrachten den Zustand $s = \langle x \mapsto ? \rangle$ (nur eine Variable):

s
-2
-1
0
1

Der Fixpunkt bei der Arbeit (I)

```
while (x < 0) {
  x= x+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} \begin{cases} \sigma & \sigma(x) \geq 0 \\ f(\sigma[\sigma(x) + 1/x]) & \sigma(x) < 0 \end{cases}$$

Wir betrachten den Zustand $s = \langle x \mapsto ? \rangle$ (nur eine Variable):

s	$\Gamma^0(s)$
-2	$\perp$
-1	$\perp$
0	$\perp$
1	$\perp$

Der Fixpunkt bei der Arbeit (I)

```
while (x < 0) {
  x= x+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} \begin{cases} \sigma & \sigma(x) \geq 0 \\ f(\sigma[\sigma(x)+1/x]) & \sigma(x) < 0 \end{cases}$$

Wir betrachten den Zustand $s = \langle x \mapsto ?\rangle$ (nur eine Variable):

s	$\Gamma^0(s)$	$\Gamma^1(s)$
-2	$\perp$	$\Gamma^0(s[-1/x]) = \perp$
-1	$\perp$	$\Gamma^0(s[0/x]) = \perp$
0	$\perp$	0
1	$\perp$	1

Der Fixpunkt bei der Arbeit (I)

```
while (x < 0) {
  x= x+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} \begin{cases} \sigma & \sigma(x) \geq 0 \\ f(\sigma[\sigma(x)+1/x]) & \sigma(x) < 0 \end{cases}$$

Wir betrachten den Zustand $s = \langle x \mapsto ? \rangle$ (nur eine Variable):

s	$\Gamma^0(s)$	$\Gamma^1(s)$	$\Gamma^2(s)$
-2	$\perp$	$\Gamma^0(s[-1/x]) = \perp$	$\Gamma^1(s[-1/x]) = \perp$
-1	$\perp$	$\Gamma^0(s[0/x]) = \perp$	$\Gamma^1(s[0/x]) = 0$
0	$\perp$	0	0
1	$\perp$	1	1

Der Fixpunkt bei der Arbeit (I)

```
while (x < 0) {
  x= x+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} \begin{cases} \sigma & \sigma(x) \geq 0 \\ f(\sigma[\sigma(x)+1/x]) & \sigma(x) < 0 \end{cases}$$

Wir betrachten den Zustand $s = \langle x \mapsto ? \rangle$ (nur eine Variable):

s	$\Gamma^0(s)$	$\Gamma^1(s)$	$\Gamma^2(s)$	$\Gamma^3(s)$
-2	$\perp$	$\Gamma^0(s[-1/x]) = \perp$	$\Gamma^1(s[-1/x]) = \perp$	$\Gamma^2(s[-1/x]) = 0$
-1	$\perp$	$\Gamma^0(s[0/x]) = \perp$	$\Gamma^1(s[0/x]) = 0$	$\Gamma^2(s[0/x]) = 0$
0	$\perp$	0	0	0
1	$\perp$	1	1	1

Der Fixpunkt bei der Arbeit (II)

```
x= 0;
while (n > 0) {
  x= x+n;
  n= n-1;
  }
```

$$\Gamma(f)(\sigma) = \begin{cases} \sigma & \sigma(n) \leq 0 \\ f(\sigma[\sigma(x) + \sigma(n)/x][\sigma(n) - 1/n]) & \sigma(n) > 0 \end{cases}$$

Wir betrachten Zustände $s = \langle x \mapsto ?, n \mapsto ? \rangle$ (zwei Variablen).
Der Wert von x im Initialzustand ist dabei unerheblich:

s
n
-1
0
1
2
3
4

Der Fixpunkt bei der Arbeit (II)

```
x= 0;
while (n > 0) {
  x= x+n;
  n= n-1;
  }
```

$$\Gamma(f)(\sigma) = \begin{cases} \sigma & \sigma(n) \leq 0 \\ f(\sigma[\sigma(x) + \sigma(n)/x][\sigma(n) - 1/n]) & \sigma(n) > 0 \end{cases}$$

Wir betrachten Zustände $s = \langle x \mapsto ?, n \mapsto ? \rangle$ (zwei Variablen).
Der Wert von x im Initialzustand ist dabei unerheblich:

s	$\Gamma^0(s)$	
n	x	n
-1	$\perp$	$\perp$
0	$\perp$	$\perp$
1	$\perp$	$\perp$
2	$\perp$	$\perp$
3	$\perp$	$\perp$
4	$\perp$	$\perp$

Der Fixpunkt bei der Arbeit (II)

```
x= 0;
while (n > 0) {
  x= x+n;
  n= n-1;
  }
```

$$\Gamma(f)(\sigma) = \begin{cases} \sigma & \sigma(n) \leq 0 \\ f(\sigma[\sigma(x) + \sigma(n)/x][\sigma(n) - 1/n]) & \sigma(n) > 0 \end{cases}$$

Wir betrachten Zustände $s = \langle x \mapsto ?, n \mapsto ? \rangle$ (zwei Variablen).
Der Wert von x im Initialzustand ist dabei unerheblich:

s	$\Gamma^0(s)$		$\Gamma^1(s)$	
n	x	n	x	n
-1	$\perp$	$\perp$	0	-1
0	$\perp$	$\perp$	0	0
1	$\perp$	$\perp$	$\perp$	$\perp$
2	$\perp$	$\perp$	$\perp$	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$
4	$\perp$	$\perp$	$\perp$	$\perp$

Der Fixpunkt bei der Arbeit (II)

```
x= 0;
while (n > 0) {
  x= x+n;
  n= n-1;
  }
```

$$\Gamma(f)(\sigma) = \begin{cases} \sigma & \sigma(n) \leq 0 \\ f(\sigma[\sigma(x)+\sigma(n)/x][\sigma(n)-1/n]) & \sigma(n) > 0 \end{cases}$$

Wir betrachten Zustände $s = \langle x \mapsto ?, n \mapsto ? \rangle$ (zwei Variablen).
Der Wert von x im Initialzustand ist dabei unerheblich:

s	$\Gamma^0(s)$		$\Gamma^1(s)$		$\Gamma^2(s)$	
n	x	n	x	n	x	n
-1	$\perp$	$\perp$	0	-1	0	-1
0	$\perp$	$\perp$	0	0	0	0
1	$\perp$	$\perp$	$\perp$	$\perp$	1	0
2	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
4	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$

Der Fixpunkt bei der Arbeit (II)

```
x= 0;
while (n > 0) {
  x= x+n;
  n= n-1;
  }
```

$$\Gamma(f)(\sigma) = \begin{cases} \sigma & \sigma(n) \leq 0 \\ f(\sigma[\sigma(x)+\sigma(n)/x][\sigma(n)-1/n]) & \sigma(n) > 0 \end{cases}$$

Wir betrachten Zustände $s = \langle x \mapsto ?, n \mapsto ? \rangle$ (zwei Variablen). Der Wert von x im Initialzustand ist dabei unerheblich:

s	$\Gamma^0(s)$		$\Gamma^1(s)$		$\Gamma^2(s)$		$\Gamma^3(s)$	
n	x	n	x	n	x	n	x	n
-1	$\perp$	$\perp$	0	-1	0	-1	0	-1
0	$\perp$	$\perp$	0	0	0	0	0	0
1	$\perp$	$\perp$	$\perp$	$\perp$	1	0	1	0
2	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	3	0
3	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
4	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$

Der Fixpunkt bei der Arbeit (II)

```
x= 0;
while (n > 0) {
  x= x+n;
  n= n-1;
  }
```

$$\Gamma(f)(\sigma) = \begin{cases} \sigma & \sigma(n) \leq 0 \\ f(\sigma[\sigma(x) + \sigma(n)/x][\sigma(n) - 1/n]) & \sigma(n) > 0 \end{cases}$$

Wir betrachten Zustände $s = \langle x \mapsto ?, n \mapsto ? \rangle$ (zwei Variablen).
Der Wert von x im Initialzustand ist dabei unerheblich:

s	$\Gamma^0(s)$		$\Gamma^1(s)$		$\Gamma^2(s)$		$\Gamma^3(s)$		$\Gamma^4(s)$	
n	x	n	x	n	x	n	x	n	x	n
-1	$\perp$	$\perp$	0	-1	0	-1	0	-1	0	-1
0	$\perp$	$\perp$	0	0	0	0	0	0	0	0
1	$\perp$	$\perp$	$\perp$	$\perp$	1	0	1	0	1	0
2	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	3	0	3	0
3	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	6	0
4	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$

Der Fixpunkt bei der Arbeit (II)

```
x= 0;
while (n > 0) {
  x= x+n;
  n= n-1;
  }
```

$$\Gamma(f)(\sigma) = \begin{cases} \sigma & \sigma(n) \leq 0 \\ f(\sigma[\sigma(x)+\sigma(n)/x][\sigma(n)-1/n]) & \sigma(n) > 0 \end{cases}$$

Wir betrachten Zustände $s = \langle x \mapsto ?, n \mapsto ? \rangle$ (zwei Variablen).
Der Wert von x im Initialzustand ist dabei unerheblich:

s	$\Gamma^0(s)$		$\Gamma^1(s)$		$\Gamma^2(s)$		$\Gamma^3(s)$		$\Gamma^4(s)$		$\Gamma^5(s)$	
n	x	n	x	n	x	n	x	n	x	n	x	n
-1	$\perp$	$\perp$	0	-1	0	-1	0	-1	0	-1	0	-1
0	$\perp$	$\perp$	0	0	0	0	0	0	0	0	0	0
1	$\perp$	$\perp$	$\perp$	$\perp$	1	0	1	0	1	0	1	0
2	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	3	0	3	0	3	0
3	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	6	0	6	0
4	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	10	0

Der Fixpunkt bei der Arbeit (III)

Kleine Änderung im Beispielprogramm:

```
x= 0;
while (n!=0) {
  x= x+n;
  n= n-1;
  }
```

$$\Gamma(f)(\sigma) = \begin{cases} \sigma & \sigma(n) = 0 \\ f(\sigma[\sigma(x) + \sigma(n)/x][\sigma(n) - 1/n]) & \text{sonst} \end{cases}$$

Jetzt ergibt sich:

s
n
-2
-1
0
1
2
3

Der Fixpunkt bei der Arbeit (III)

Kleine Änderung im Beispielprogramm:

```
x= 0;
while (n!=0) {
  x= x+n;
  n= n-1;
  }
```

$$\Gamma(f)(\sigma) = \begin{cases} \sigma & \sigma(n) = 0 \\ f(\sigma[\sigma(x) + \sigma(n)/x][\sigma(n) - 1/n]) & \text{sonst} \end{cases}$$

Jetzt ergibt sich:

s	$\Gamma^0(s)$	
n	x	n
-2	$\perp$	$\perp$
-1	$\perp$	$\perp$
0	$\perp$	$\perp$
1	$\perp$	$\perp$
2	$\perp$	$\perp$
3	$\perp$	$\perp$

Der Fixpunkt bei der Arbeit (III)

Kleine Änderung im Beispielprogramm:

```
x= 0;
while (n!=0) {
  x= x+n;
  n= n-1;
  }
```

$$\Gamma(f)(\sigma) = \begin{cases} \sigma & \sigma(n) = 0 \\ f(\sigma[\sigma(x) + \sigma(n)/x][\sigma(n) - 1/n]) & \text{sonst} \end{cases}$$

Jetzt ergibt sich:

s	$\Gamma^0(s)$		$\Gamma^1(s)$	
n	x	n	x	n
-2	$\perp$	$\perp$	$\perp$	$\perp$
-1	$\perp$	$\perp$	$\perp$	$\perp$
0	$\perp$	$\perp$	0	0
1	$\perp$	$\perp$	$\perp$	$\perp$
2	$\perp$	$\perp$	$\perp$	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$

Der Fixpunkt bei der Arbeit (III)

Kleine Änderung im Beispielprogramm:

```
x= 0;
while (n!=0) {
  x= x+n;
  n= n-1;
  }
```

$$\Gamma(f)(\sigma) = \begin{cases} \sigma & \sigma(n) = 0 \\ f(\sigma[\sigma(x) + \sigma(n)/x][\sigma(n) - 1/n]) & \text{sonst} \end{cases}$$

Jetzt ergibt sich:

s	$\Gamma^0(s)$		$\Gamma^1(s)$		$\Gamma^2(s)$	
n	x	n	x	n	x	n
-2	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
-1	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
0	$\perp$	$\perp$	0	0	0	0
1	$\perp$	$\perp$	$\perp$	$\perp$	1	0
2	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$

Der Fixpunkt bei der Arbeit (III)

Kleine Änderung im Beispielprogramm:

```
x= 0;
while (n!=0) {
  x= x+n;
  n= n-1;
  }
```

$$\Gamma(f)(\sigma) = \begin{cases} \sigma & \sigma(n) = 0 \\ f(\sigma[\sigma(x) + \sigma(n)/x][\sigma(n) - 1/n]) & \text{sonst} \end{cases}$$

Jetzt ergibt sich:

s	$\Gamma^0(s)$		$\Gamma^1(s)$		$\Gamma^2(s)$		$\Gamma^3(s)$	
n	x	n	x	n	x	n	x	n
-2	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
-1	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
0	$\perp$	$\perp$	0	0	0	0	0	0
1	$\perp$	$\perp$	$\perp$	$\perp$	1	0	1	0
2	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	3	0
3	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$

Der Fixpunkt bei der Arbeit (III)

Kleine Änderung im Beispielprogramm:

```
x= 0;
while (n!=0) {
  x= x+n;
  n= n-1;
  }
```

$$\Gamma(f)(\sigma) = \begin{cases} \sigma & \sigma(n) = 0 \\ f(\sigma[\sigma(x) + \sigma(n)/x][\sigma(n) - 1/n]) & \text{sonst} \end{cases}$$

Jetzt ergibt sich:

s	$\Gamma^0(s)$		$\Gamma^1(s)$		$\Gamma^2(s)$		$\Gamma^3(s)$		$\Gamma^4(s)$	
n	x	n	x	n	x	n	x	n	x	n
-2	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
-1	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
0	$\perp$	$\perp$	0	0	0	0	0	0	0	0
1	$\perp$	$\perp$	$\perp$	$\perp$	1	0	1	0	1	0
2	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	3	0	3	0
3	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	6	0

Der Fixpunkt bei der Arbeit (IV)

```
while (1) {
  x= x+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} f(\sigma[\sigma(x) + 1/x])$$

Jetzt ergibt sich:

s

-2

-1

0

1

2

3

Der Fixpunkt bei der Arbeit (IV)

```
while (1) {
  x= x+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} f(\sigma[\sigma(x) + 1/x])$$

Jetzt ergibt sich:

s	$\Gamma^0(s)$
-2	$\perp$
-1	$\perp$
0	$\perp$
1	$\perp$
2	$\perp$
3	$\perp$

Der Fixpunkt bei der Arbeit (IV)

```
while (1) {
  x= x+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} f(\sigma[\sigma(x) + 1/x])$$

Jetzt ergibt sich:

s	$\Gamma^0(s)$	$\Gamma^1(s)$
-2	$\perp$	$\perp$
-1	$\perp$	$\perp$
0	$\perp$	$\perp$
1	$\perp$	$\perp$
2	$\perp$	$\perp$
3	$\perp$	$\perp$

Der Fixpunkt bei der Arbeit (IV)

```
while (1) {
  x= x+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} f(\sigma[\sigma(x) + 1/x])$$

Jetzt ergibt sich:

s	$\Gamma^0(s)$	$\Gamma^1(s)$	$\Gamma^2(s)$
-2	$\perp$	$\perp$	$\perp$
-1	$\perp$	$\perp$	$\perp$
0	$\perp$	$\perp$	$\perp$
1	$\perp$	$\perp$	$\perp$
2	$\perp$	$\perp$	$\perp$
3	$\perp$	$\perp$	$\perp$

Der Fixpunkt bei der Arbeit (IV)

```
while (1) {
  x= x+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} f(\sigma[\sigma(x) + 1/x])$$

Jetzt ergibt sich:

s	$\Gamma^0(s)$	$\Gamma^1(s)$	$\Gamma^2(s)$	$\Gamma^3(s)$
-2	$\perp$	$\perp$	$\perp$	$\perp$
-1	$\perp$	$\perp$	$\perp$	$\perp$
0	$\perp$	$\perp$	$\perp$	$\perp$
1	$\perp$	$\perp$	$\perp$	$\perp$
2	$\perp$	$\perp$	$\perp$	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$

Arbeitsblatt 3.4: Semantik III

Wir betrachten das Beispielprogramm:

```
x= 1;
while (n > 0) {
  x= x*n;
  n= n-1;
  }
```

Berechnen Sie wie oben den Fixpunkt:

```
s    G^0        G^1          G^2        G^3       G^4
n   x    n     x    n       x    n     x    n    x    n
0
1
2
3
```

Der Fixpunkt bei der Arbeit (V)

```
x= 0;
i= 0;
while (i<=n) {
  x= x+i;
  i= i+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} \begin{cases} \sigma & \sigma(i) > \sigma(n) \\ f(\sigma[\sigma(x)+\sigma(i)/x][\sigma(i)+1/i]) & \text{sonst} \end{cases}$$

Wir betrachten nur die **while**-Schleife
mit $s = \langle n \mapsto ?, i \mapsto ?, x \mapsto ? \rangle$.

s	
n	i
0	0
0	1
1	0
1	1
1	2
2	0
2	1
2	2
2	3

Der Fixpunkt bei der Arbeit (V)

```
x= 0;
i= 0;
while (i<=n) {
  x= x+i;
  i= i+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} \begin{cases} \sigma & \sigma(i) > \sigma(n) \\ f(\sigma[\sigma(x)+\sigma(i)/x][\sigma(i)+1/i]) & \text{sonst} \end{cases}$$

Wir betrachten nur die **while**-Schleife mit $s = \langle n \mapsto ?, i \mapsto ?, x \mapsto ? \rangle$.

s		$\Gamma^0(s)$		
n	i	n	i	x
0	0	$\perp$	$\perp$	$\perp$
0	1	$\perp$	$\perp$	$\perp$
1	0	$\perp$	$\perp$	$\perp$
1	1	$\perp$	$\perp$	$\perp$
1	2	$\perp$	$\perp$	$\perp$
2	0	$\perp$	$\perp$	$\perp$
2	1	$\perp$	$\perp$	$\perp$
2	2	$\perp$	$\perp$	$\perp$
2	3	$\perp$	$\perp$	$\perp$

Der Fixpunkt bei der Arbeit (V)

```
x= 0;
i= 0;
while (i<=n) {
  x= x+i;
  i= i+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} \begin{cases} \sigma & \sigma(i) > \sigma(n) \\ f(\sigma[\sigma(x)+\sigma(i)/x][\sigma(i)+1/i]) & \text{sonst} \end{cases}$$

Wir betrachten nur die **while**-Schleife mit $s = \langle n \mapsto ?, i \mapsto ?, x \mapsto ? \rangle$.

s		$\Gamma^1(s)$		
n	i	n	i	x
0	0	$\perp$	$\perp$	$\perp$
0	1	0	1	x
1	0	$\perp$	$\perp$	$\perp$
1	1	$\perp$	$\perp$	$\perp$
1	2	1	2	x
2	0	$\perp$	$\perp$	$\perp$
2	1	$\perp$	$\perp$	$\perp$
2	2	$\perp$	$\perp$	$\perp$
2	3	2	3	x

Der Fixpunkt bei der Arbeit (V)

```
x=  0;
i=  0;
while  (i<=n)  {
  x=  x+i ;
  i=  i+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} \begin{cases} \sigma & \sigma(i) > \sigma(n) \\ f(\sigma[\sigma(x)+\sigma(i)/x][\sigma(i)+1/i]) & \text{sonst} \end{cases}$$

Wir betrachten nur die **while**-Schleife mit $s = \langle n \mapsto ?, i \mapsto ?, x \mapsto ? \rangle$.

s		$\Gamma^1(s)$			$\Gamma^2(s)$		
n	i	n	i	x	n	i	x
0	0	$\perp$	$\perp$	$\perp$	0	1	x
0	1	0	1	x	0	1	x
1	0	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
1	1	$\perp$	$\perp$	$\perp$	1	2	$x+1$
1	2	1	2	x	1	2	x
2	0	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
2	1	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
2	2	$\perp$	$\perp$	$\perp$	2	3	$x+2$
2	3	2	3	x	2	3	x

Der Fixpunkt bei der Arbeit (V)

```
x= 0;
i= 0;
while (i<=n) {
  x= x+i;
  i= i+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} \begin{cases} \sigma & \sigma(i) > \sigma(n) \\ f(\sigma[\sigma(x)+\sigma(i)/x][\sigma(i)+1/i]) & \text{sonst} \end{cases}$$

Wir betrachten nur die **while**-Schleife
mit $s = \langle n \mapsto ?, i \mapsto ?, x \mapsto ? \rangle$.

s		$\Gamma^1(s)$			$\Gamma^2(s)$			$\Gamma^3(s)$		
n	i	n	i	x	n	i	x	n	i	x
0	0	$\bot$	$\bot$	$\bot$	0	1	x	0	1	x
0	1	0	1	x	0	1	x	0	1	x
1	0	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	1	2	$x+1$
1	1	$\bot$	$\bot$	$\bot$	1	2	$x+1$	1	2	$x+1$
1	2	1	2	x	1	2	x	1	2	x
2	0	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
2	1	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	2	3	$x+3$
2	2	$\bot$	$\bot$	$\bot$	2	3	$x+2$	2	3	$x+2$
2	3	2	3	x	2	3	x	2	3	x

Der Fixpunkt bei der Arbeit (V)

```
x= 0;
i= 0;
while (i<=n) {
  x= x+i;
  i= i+1;
  }
```

$$\Gamma(f)(\sigma) \stackrel{def}{=} \begin{cases} \sigma & \sigma(i) > \sigma(n) \\ f(\sigma[\sigma(x)+\sigma(i)/x][\sigma(i)+1/i]) & \text{sonst} \end{cases}$$

Wir betrachten nur die **while**-Schleife
mit $s = \langle n \mapsto ?, i \mapsto ?, x \mapsto ? \rangle$.

s		$\Gamma^1(s)$			$\Gamma^2(s)$			$\Gamma^3(s)$			$\Gamma^4(s)$		
n	i	n	i	x	n	i	x	n	i	x	n	i	x
0	0	$\perp$	$\perp$	$\perp$	0	1	x	0	1	x	0	1	x
0	1	0	1	x	0	1	x	0	1	x	0	1	x
1	0	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	1	2	$x+1$	1	2	$x+1$
1	1	$\perp$	$\perp$	$\perp$	1	2	$x+1$	1	2	$x+1$	1	2	$x+1$
1	2	1	2	x	1	2	x	1	2	x	1	2	x
2	0	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	2	3	$x+3$
2	1	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	2	3	$x+3$	2	3	$x+3$
2	2	$\perp$	$\perp$	$\perp$	2	3	$x+2$	2	3	$x+2$	2	3	$x+2$
2	3	2	3	x	2	3	x	2	3	x	2	3	x

Weitere Eigenschaften der denotationalen Semantik

Lemma (Partielle Funktion)

$[\![-]\!]_{\mathcal{C}}$ *ist rechtseindeutig und damit eine* ***partielle Funktion****.*

- Beweis über strukturelle Induktion über $c \in \textbf{Stmt}$ und über **Fixpunktinduktion**:
  - Zu zeigen: wenn s rechtseindeutig, dann ist $\Gamma(s)$ rechtseindeutig
  - Dann ist $\mathit{fix}(\Gamma)$ rechtseindeutig.
- Eigenschaften der Iteration:
  - Sei $w \equiv \textbf{while}\ (b)\ c$
  - Dann

$$[\![w]\!]_{\mathcal{C}} = [\![\textbf{if}\ (b)\ \{c; w\}\ \textbf{else}\ \{\ \}]\!]_{\mathcal{C}} \tag{1}$$

$$(\sigma, \sigma') \in [\![w]\!]_{\mathcal{C}} \Longrightarrow (\sigma', \mathit{false}) \in [\![b]\!]_{\mathcal{B}} \tag{2}$$

Beweis (1)

Zu zeigen: $[\![w]\!]_{\mathcal{C}} = [\![\textbf{if}\ (b)\ \{c; w\}\ \textbf{else}\ \{\,\}]\!]_{\mathcal{C}}$

$$
\begin{aligned}
[\![\text{if } (b)\ \{c;w\} \text{ else } \{\}]\!]_{\mathcal{C}} &= \{(\sigma,\sigma') \mid (\sigma,\text{true}) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma,\sigma') \in [\![c;w]\!]_{\mathcal{C}}\} \\
&\quad \cup \{(\sigma,\sigma') \mid (\sigma,\text{false}) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma,\sigma') \in [\![\{\}]\!]_{\mathcal{C}}\} \\
&= \{(\sigma,\sigma') \mid (\sigma,\text{true}) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma,\sigma') \in [\![c]\!]_{\mathcal{C}} \circ [\![w]\!]_{\mathcal{C}}\} \\
&\quad \cup \{(\sigma,\sigma') \mid (\sigma,\text{false}) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma,\sigma') \in Id_{\Sigma}\} \\
&= \{(\sigma,\sigma') \mid (\sigma,\text{true}) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma,\sigma') \in [\![c]\!]_{\mathcal{C}} \circ [\![w]\!]_{\mathcal{C}}\} \\
&\quad \cup \{(\sigma,\sigma) \mid (\sigma,\text{false}) \in [\![b]\!]_{\mathcal{B}}\} \\
&= \Gamma([\![w]\!]_{\mathcal{C}}) \\
&= \Gamma(\text{fix}(\Gamma)) = \text{fix}(\Gamma) = [\![w]\!]_{\mathcal{C}} \qquad \square
\end{aligned}
$$

Zusammenfassung

- Die denotationale Semantik bildet Programme (Ausdrücke) auf **partielle Funktionen** $\Sigma \rightharpoonup \Sigma$ ab.
- Zentral ist der Begriff des **kleinsten Fixpunktes**, der die Semantik der while-Schleife bildet.
- Undefiniertheit wird **implizit** behandelt (durch die Partialität von $\Sigma \rightharpoonup \Sigma$).
  - Nicht-Termination und Undefiniertheit sind semantisch äquivalent.
- Genaues Verhältnis zur **operationalen Semantik?** Nächste Vorlesung