Korrekte Software: Grundlagen und Methoden

Vorlesung 3 vom 17.04.24

Denotationale Semantik

Serge Autexier, Christoph Lüth

Universität Bremen

Sommersemester 2024

Fahrplan

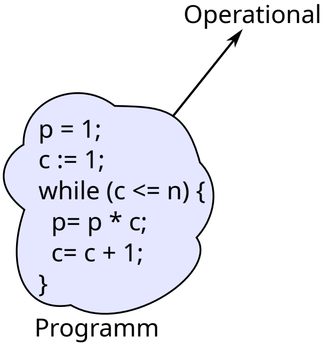
- Einführung
- Operationale Semantik
- Denotationale Semantik
- Äquivalenz der Operationalen und Denotationalen Semantik
- Der Floyd-Hoare-Kalkül
- Invarianten im Floyd-Hoare-Kalkül
- Korrektheit des Floyd-Hoare-Kalküls
- Strukturierte Datentypen
- Verifikationsbedingungen
- Funktionen und Prozeduren I
- Funktionen und Prozeduren II
- Referenzen
- Ausblick und Rückblick

Überblick

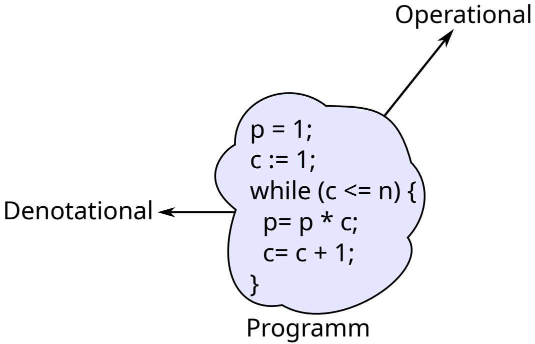
```
p = 1;
c := 1;
while (c <= n) {
  p= p * c;
  c= c + 1;
}
```

Programm

Überblick



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 überführen:

  $$\langle c, \sigma \rangle \rightarrow_{Stmt} \sigma'$$

- **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 — 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 Funktionen und ihre Graphen

- Der **Graph** einer partiellen Funktion $f : X \rightharpoonup Y$ ist eine Relation

$$grph(f) \subseteq X \times Y \overset{def}{=} \{(x, f(x)) \mid x \in \mathrm{dom}(f)\}$$

- Wir können eine partielle Funktion durch ihren Graph definieren:

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**)

- Wir benutzen beide Notationen, aber für die denotationale Semantik die Graph-Notation.
- **Systemzustände** sind partielle Abbildungen $\Sigma \overset{def}{=} \mathbf{Loc} \rightharpoonup \mathbf{V}$ ($\longrightarrow$ letzte VL)

Beispiel

Als Beispiel betrachten wir die partielle Funktion $\mathit{div3} : \{0 \ldots 10\} \to \mathbb{N}$

$$\mathit{div3}(x) = y \quad \text{g.d.w.} \quad 3 \cdot y = x$$

- Zuordnung:

$$\begin{array}{ccc} 0 & \mapsto & 0 \\ 1 & & \\ 2 & & \\ 3 & \mapsto & 1 \\ 4 & & \\ 5 & & \\ 6 & \mapsto & 2 \\ 7 & & \\ 8 & & \\ 9 & \mapsto & 3 \\ 10 & & \end{array}$$

- Notation als Relation (**Graph**):

$$\mathit{div3} \stackrel{\mathit{def}}{=} \{(0,0), (3,1), (6,2), (9,3)\}$$

- Wir schreiben

$$\begin{array}{ll} \mathit{div3}(3) = 1 & \text{für } (3,1) \in \mathit{div3} \\ \mathit{div3}(5) = \perp & \text{für es gibt kein } y \text{ mit } (5,y) \in \mathit{div3} \\ \mathit{div3}(5) = \perp & \text{für } \forall y. (5,y) \notin \mathit{div3} \end{array}$$

- Achtung, Partialität!

Achtung, Partialität!

- Beim Rechnen mit partiellen Funktionen muss die **Definiertheit** beachtet werden.
- Insbesondere darf nicht mit undefinierten Ausdrücken gerechnet werden.
- Bspw. gilt oben **nicht** im allgemeinen:

$$3 \cdot \mathit{div3}(x) = x \quad \times$$

oder

$$\mathit{div3}(1) = \perp = \mathit{div3}(2) \Longrightarrow \mathit{div3}(1) = \mathit{div3}(2) \quad \times$$

- Warum? Dann gälte

Achtung, Partialität!

- Beim Rechnen mit partiellen Funktionen muss die **Definiertheit** beachtet werden.
- Insbesondere darf nicht mit undefinierten Ausdrücken gerechnet werden.
- Bspw. gilt oben **nicht** im allgemeinen:

$$3 \cdot \mathit{div3}(x) = x \quad \times$$

oder

$$\mathit{div3}(1) = \perp = \mathit{div3}(2) \Longrightarrow \mathit{div3}(1) = \mathit{div3}(2) \quad \times$$

- Warum? Dann gälte

$$\mathit{div3}(1) = \mathit{div3}(2)$$

Achtung, Partialität!

- Beim Rechnen mit partiellen Funktionen muss die **Definiertheit** beachtet werden.
- Insbesondere darf nicht mit undefinierten Ausdrücken gerechnet werden.
- Bspw. gilt oben **nicht** im allgemeinen:

$$3 \cdot \mathit{div3}(x) = x \quad \times$$

oder

$$\mathit{div3}(1) = \perp = \mathit{div3}(2) \Longrightarrow \mathit{div3}(1) = \mathit{div3}(2) \quad \times$$

- Warum? Dann gälte

$$\begin{aligned} \mathit{div3}(1) &= \mathit{div3}(2) \\ 3 \cdot \mathit{div3}(1) &= 3 \cdot \mathit{div3}(2) \end{aligned}$$

Achtung, Partialität!

- Beim Rechnen mit partiellen Funktionen muss die **Definiertheit** beachtet werden.
- Insbesondere darf nicht mit undefinierten Ausdrücken gerechnet werden.
- Bspw. gilt oben **nicht** im allgemeinen:

$$3 \cdot \mathit{div3}(x) = x \quad \times$$

oder

$$\mathit{div3}(1) = \bot = \mathit{div3}(2) \Longrightarrow \mathit{div3}(1) = \mathit{div3}(2) \quad \times$$

- Warum? Dann gälte

$$\mathit{div3}(1) = \mathit{div3}(2)$$
$$3 \cdot \mathit{div3}(1) = 3 \cdot \mathit{div3}(2)$$
$$1 = 2 \quad \lightning$$

Achtung, Partialität!

- Beim Rechnen mit partiellen Funktionen muss die **Definiertheit** beachtet werden.
- Insbesondere darf nicht mit undefinierten Ausdrücken gerechnet werden.
- Bspw. gilt oben **nicht** im allgemeinen:

$$3 \cdot \mathit{div3}(x) = x \quad \times$$

oder

$$\mathit{div3}(1) = \perp = \mathit{div3}(2) \Longrightarrow \mathit{div3}(1) = \mathit{div3}(2) \quad \times$$

- Warum? Dann gälte

$$\begin{aligned} \mathit{div3}(1) &= \mathit{div3}(2) \\ 3 \cdot \mathit{div3}(1) &= 3 \cdot \mathit{div3}(2) \\ 1 &= 2 \quad ↯ \end{aligned}$$

- Vgl. `https://de.wikipedia.org/wiki/Trugschluss_(Mathematik)#Division_durch_0`

Arbeitsblatt 3.1: Relationen als Funktionen

Definiert wie im Beispiel eben die Funktion $sqrt : \{0, \ldots, 100\} \to \mathbb{N}$ mit

$$sqrt(x) = y \quad \text{g.d.w.} \quad y^2 = x$$

Was ist der Wert folgender Ausdrücke:

$$t_1 = 5 - sqrt(32) \quad t_2 = sqrt(49) + sqrt(0) \quad t_3 = \sqrt{3} \cdot sqrt(3) \quad t_4 = \frac{sqrt(64)}{0}$$

Arbeitsblatt 3.1: Relationen als Funktionen

Definiert wie im Beispiel eben die Funktion $sqrt : \{0, \ldots, 100\} \to \mathbb{N}$ mit

$$sqrt(x) = y \quad \text{g.d.w.} \quad y^2 = x$$

Was ist der Wert folgender Ausdrücke:

$$t_1 = 5 - sqrt(32) \quad t_2 = sqrt(49) + sqrt(0) \quad t_3 = \sqrt{3} \cdot sqrt(3) \quad t_4 = \frac{sqrt(64)}{0}$$

Denotierende Funktionen (Denotate)

- Arithmetische Ausdrücke: $a \in \mathbf{Aexp}$ denotieren eine partielle Funktion $\Sigma \rightharpoonup \mathbb{Z}$
- Boolsche Ausdrücke: $b \in \mathbf{Bexp}$ denotieren eine partielle Funktion $\Sigma \rightharpoonup \mathbb{B}$
- Anweisungen: $c \in \mathbf{Stmt}$ denotieren eine partielle Funktion $\Sigma \rightharpoonup \Sigma$

Denotat von Aexp

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

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

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 $v_1 = \sigma(x) = v_2$.
- Induktionsschritt sind die anderen Klauseln.
  Sei $a \equiv a_1 + a_2$.
  Induktionsannahme ist: wenn $(\sigma, n_i) \in [\![a_i]\!]_{\mathcal{A}}, (\sigma, m_i) \in [\![a_i]\!]_{\mathcal{A}}$ dann $n_i = m_i$.
  Sei $v_1 = n_1 + n_2$ mit $(\sigma, n_1) \in [\![a_1]\!]_{\mathcal{A}}, (\sigma, n_2) \in [\![a_2]\!]_{\mathcal{A}}$, und $v_2 = m_1 + m_2$ mit $(\sigma, m_1) \in [\![a_1]\!]_{\mathcal{A}}, (\sigma, m_2) \in [\![a_2]\!]_{\mathcal{A}}$.

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 $v_1 = \sigma(x) = v_2$.
- Induktionsschritt sind die anderen Klauseln.
  Sei $a \equiv a_1 + a_2$.
  Induktionsannahme ist: wenn $(\sigma, n_i) \in [\![a_i]\!]_{\mathcal{A}}, (\sigma, m_i) \in [\![a_i]\!]_{\mathcal{A}}$ dann $n_i = m_i$.
  Sei $v_1 = n_1 + n_2$ mit $(\sigma, n_1) \in [\![a_1]\!]_{\mathcal{A}}, (\sigma, n_2) \in [\![a_2]\!]_{\mathcal{A}}$, und $v_2 = m_1 + m_2$ mit $(\sigma, m_1) \in [\![a_1]\!]_{\mathcal{A}}, (\sigma, m_2) \in [\![a_2]\!]_{\mathcal{A}}$.
  Aus der Annahme folgt $n_1 = m_1$ und $n_2 = m_2$, 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**:

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

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

Arbeitsblatt 3.2: 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?

Lösung

Denotat von Bexp

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

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

Denotat von Bexp

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

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

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:

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

Arbeitsblatt 3.3: Semantik II

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

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

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

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

Lösung

Denotationale Semantik von Anweisungen

- Zuweisung: punktweise Änderung des Zustands σ zu $\sigma[x \mapsto n]$
- 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: punktweise Änderung des Zustands σ zu $\sigma[x \mapsto n]$
- 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,

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

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

Arbeitsblatt 3.4: 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 $R \circ S = \{(1,?), \ldots\}$

Arbeitsblatt 3.4: 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 $R \circ S = \{(1,?), \ldots\}$

Arbeitsblatt 3.4: Komposition von Relationen

Zur Übung: betrachten Sie folgende Relationen:

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

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

Denotat von Stmt

$$\llbracket . \rrbracket_{\mathcal{C}} : \textbf{Stmt} \to (\Sigma \rightharpoonup \Sigma)$$

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

Denotat von Stmt

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

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

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

$$\begin{aligned}
[\![w]\!]_{\mathcal{C}} &\stackrel{?}{=} [\![\textbf{if}\ (b)\ \{c; w\}\ \textbf{else}\ \{\,\}]\!]_{\mathcal{C}} \\
&= \{(\sigma, \sigma') \mid (\sigma, \mathit{true}) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma, \sigma') \in [\![c]\!]_{\mathcal{C}} \circ [\![w]\!]_{\mathcal{C}}\} \\
&\quad \cup \{(\sigma, \sigma') \mid (\sigma, \mathit{false}) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma, \sigma') \in [\![\{\,\}]\!]_{\mathcal{C}}\}
\end{aligned}$$

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

$$x = F(x)$$

- Das ist ein **Fixpunkt**:

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

- Was ist das?

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

$$\begin{aligned}
[\![w]\!]_{\mathcal{C}} &\stackrel{?}{=} [\![\textbf{if}\ (b)\ \{c; w\}\ \textbf{else}\ \{\,\}]\!]_{\mathcal{C}} \\
&= \{(\sigma, \sigma') \mid (\sigma, \mathit{true}) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma, \sigma') \in [\![c]\!]_{\mathcal{C}} \circ [\![w]\!]_{\mathcal{C}}\} \\
&\quad \cup \{(\sigma, \sigma') \mid (\sigma, \mathit{false}) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma, \sigma') \in [\![\{\,\}]\!]_{\mathcal{C}}\}
\end{aligned}$$

- 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.

Denotationale Semantik für die Iteration

- Sei $w \equiv$ **while** $(b)\ c$
- Konstruktion: "Auffalten" der Schleife (f ist ein Denotat):

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

- b und c sind Parameter von Γ
- Dann ist

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

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)

Denotation für Stmt

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

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

Der Fixpunkt bei der Arbeit (I)

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

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

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

Der Fixpunkt bei der Arbeit (I)

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

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

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

$$\begin{array}{c} s \\ -2 \\ -1 \\ 0 \\ 1 \end{array}$$

Der Fixpunkt bei der Arbeit (I)

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

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

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;
  }
```

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

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

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

Der Fixpunkt bei der Arbeit (I)

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

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

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

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

Der Fixpunkt bei der Arbeit (I)

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

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

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	$\perp$	$\Gamma^0(s[x \mapsto -1]) = \perp$	$\Gamma^1(s[x \mapsto -1]) = \perp$	$\Gamma^2(s[x \mapsto -1]) = 0$
-1	$\perp$	$\Gamma^0(s[x \mapsto 0]) = \perp$	$\Gamma^1(s[x \mapsto 0]) = 0$	$\Gamma^2(s[x \mapsto 0]) = 0$
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;
  }
```

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

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;
  }
```

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

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;
  }
```

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

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;
  }
```

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

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;
  }
```

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

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;
  }
```

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

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;
  }
```

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

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	$\bot$	$\bot$	0	-1	0	-1	0	-1	0	-1	0	-1
0	$\bot$	$\bot$	0	0	0	0	0	0	0	0	0	0
1	$\bot$	$\bot$	$\bot$	$\bot$	1	0	1	0	1	0	1	0
2	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	3	0	3	0	3	0
3	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	6	0	6	0
4	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	10	0

Der Fixpunkt bei der Arbeit (III)

Kleine Änderung im Beispielprogramm:

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

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

Jetzt ergibt sich:

$$\begin{array}{r} s \\ n \\ -2 \\ -1 \\ 0 \\ 1 \\ 2 \\ 3 \end{array}$$

Der Fixpunkt bei der Arbeit (III)

Kleine Änderung im Beispielprogramm:

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

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

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;
  }
```

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

Jetzt ergibt sich:

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

Der Fixpunkt bei der Arbeit (III)

Kleine Änderung im Beispielprogramm:

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

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

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;
  }
```

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

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;
  }
```

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

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	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
-1	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
0	$\bot$	$\bot$	0	0	0	0	0	0	0	0
1	$\bot$	$\bot$	$\bot$	$\bot$	1	0	1	0	1	0
2	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	3	0	3	0
3	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	6	0

Der Fixpunkt bei der Arbeit (IV)

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

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

Jetzt ergibt sich:

s

-2

-1

0

1

2

3

Der Fixpunkt bei der Arbeit (IV)

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

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

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;
  }
```

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

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;
  }
```

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

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;
  }
```

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

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 3.5: 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										

Arbeitsblatt 3.5: Semantik III

Wir betrachten das Beispielprogramm:

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

Der Fixpunkt bei der Arbeit (V)

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

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

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;
  }
```

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

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;
  }
```

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

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

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

Der Fixpunkt bei der Arbeit (V)

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

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

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

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

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

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

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

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

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

s		$\Gamma^0(s)$			$\Gamma^1(s)$			$\Gamma^2(s)$			$\Gamma^3(s)$			$\Gamma^4(s)$		
n	i	n	i	x	n	i	x	n	i	x	n	i	x	n	i	x
0	0	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	0	1	x	0	1	x	0	1	x
0	1	$\bot$	$\bot$	$\bot$	0	1	x	0	1	x	0	1	x	0	1	x
1	0	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	1	2	$x+1$	1	2	$x+1$
1	1	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	1	2	$x+1$	1	2	$x+1$	1	2	$x+1$
1	2	$\bot$	$\bot$	$\bot$	1	2	x	1	2	x	1	2	x	1	2	x
2	0	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	2	3	$x+3$
2	1	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	2	3	$x+3$	2	3	$x+3$
2	2	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	2	3	$x+2$	2	3	$x+2$	2	3	$x+2$
2	3	$\bot$	$\bot$	$\bot$	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 $c \in$ **Stmt** und über **Fixpunktinduktion**:
  - Zu zeigen: wenn s rechtseindeutig, dann ist $\Gamma(s)$ rechtseindeutig
  - Dann ist $fix(\Gamma)$ rechtseindeutig.
- Eigenschaften der Iteration:
  - Sei $w \equiv$ **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)

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

Note

$$\mathit{fix}(\Gamma) = \Gamma(\mathit{fix}(\Gamma))$$

Beweis (1)

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

Note

$$\mathit{fix}(\Gamma) = \Gamma(\mathit{fix}(\Gamma))$$

Beweis (1)

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

Note

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

Beweis (1)

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

Note

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

Beweis (1)

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

Note

Beweis (1)

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

Note

Beweis (1)

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

Note

$$
\begin{aligned}
[\![\textbf{if}\ (b)\ c_0\ \ \textbf{else}\ \ c_1]\!]_{\mathcal{C}} = \ & \{(\sigma, \sigma') \mid (\sigma, true) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma, \sigma') \in [\![c_0]\!]_{\mathcal{C}}\} \\
& \cup \{(\sigma, \sigma') \mid (\sigma, false) \in [\![b]\!]_{\mathcal{B}} \wedge (\sigma, \sigma') \in [\![c_1]\!]_{\mathcal{C}}\}
\end{aligned}
$$

Beweis (1)

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

Note

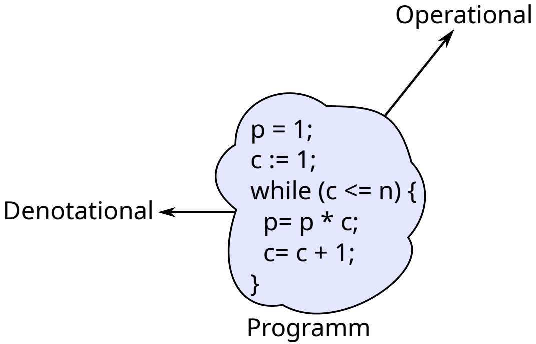
$$
\begin{aligned}
[\![\textbf{if}\ (b)\ c_0\ \textbf{else}\ c_1]\!]_{\mathcal{C}} = \ & \{(\sigma, \sigma') \mid (\sigma, \mathit{true}) \in [\![b]\!]_{\mathcal{B}} \land (\sigma, \sigma') \in [\![c_0]\!]_{\mathcal{C}}\} \\
& \cup \{(\sigma, \sigma') \mid (\sigma, \mathit{false}) \in [\![b]\!]_{\mathcal{B}} \land (\sigma, \sigma') \in [\![c_1]\!]_{\mathcal{C}}\}
\end{aligned}
$$

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?**

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



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