Semantik von Programmiersprachen

Alexander Knapp

Ludwig-Maximilians-Universität München

IMP: Syntaktische Kategorien

$$
\begin{aligned}
n &\in \mathrm{Num} \\
x &\in \mathrm{Var}
\end{aligned}
$$

$$
\begin{aligned}
a \in \mathrm{AExp} ::= {} & n \mid x \\
& \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \\
b \in \mathrm{BExp} ::= {} & \texttt{true} \mid \texttt{false} \\
& \mid a_1 = a_2 \mid a_1 \texttt{<=} a_2 \\
& \mid \texttt{not}\ b \mid b_1\ \texttt{and}\ b_2
\end{aligned}
$$

$$
\begin{aligned}
S \in \mathrm{Stm} ::= {} & \texttt{skip} \\
& \mid x \texttt{:=} a \\
& \mid S_1\ \texttt{;}\ S_2 \\
& \mid \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2 \\
& \mid \texttt{while}\ b\ \texttt{do}\ S
\end{aligned}
$$

IMP: Semantische Kategorien

- Ganze Zahlen $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$
  - Wert von n $\mathcal{N}[\![n]\!]$
- Wahrheitswerte $\mathbb{B} = \{tt, ff\}$
- Zustände Σ Funktionen $\sigma : \mathrm{Var} \to \mathbb{Z}$
  - Variablenzugriff $\sigma(x)$
  - Variablenupdate $\sigma[x \mapsto v]$

$$\sigma[x \mapsto v](x') = \begin{cases} v, & \text{falls } x = x' \\ \sigma(x'), & \text{falls } x \neq x' \end{cases}$$

IMP: Kompositionale Semantik von Ausdrücken

Semantische Funktion $\mathcal{A}[\![-]\!] : \mathrm{AExp} \to (\Sigma \to \mathbb{Z})$

$$\mathcal{A}[\![n]\!]\,\sigma = \mathcal{N}[\![n]\!]$$
$$\mathcal{A}[\![x]\!]\,\sigma = \sigma(x)$$

$$\mathcal{A}[\![a_1 + a_2]\!]\,\sigma = \mathcal{A}[\![a_1]\!]\,\sigma + \mathcal{A}[\![a_2]\!]\,\sigma$$
$$\mathcal{A}[\![a_1 - a_2]\!]\,\sigma = \mathcal{A}[\![a_1]\!]\,\sigma - \mathcal{A}[\![a_2]\!]\,\sigma$$
$$\mathcal{A}[\![a_1 * a_2]\!]\,\sigma = \mathcal{A}[\![a_1]\!]\,\sigma \cdot \mathcal{A}[\![a_2]\!]\,\sigma$$

IMP: Kompositionale Semantik von Ausdrücken

Semantische Funktion $\mathcal{B}[\![-]\!] : \mathrm{BExp} \to (\Sigma \to \mathbb{B})$

$$\mathcal{B}[\![\texttt{true}]\!]\,\sigma = tt$$
$$\mathcal{B}[\![\texttt{false}]\!]\,\sigma = ff$$

$$\mathcal{B}[\![a_1 = a_2]\!]\,\sigma = \begin{cases} tt, & \text{falls } \mathcal{A}[\![a_1]\!]\,\sigma = \mathcal{A}[\![a_2]\!]\,\sigma \\ ff, & \text{falls } \mathcal{A}[\![a_1]\!]\,\sigma \neq \mathcal{A}[\![a_2]\!]\,\sigma \end{cases}$$

$$\mathcal{B}[\![a_1 \texttt{<=} a_2]\!]\,\sigma = \begin{cases} tt, & \text{falls } \mathcal{A}[\![a_1]\!]\,\sigma \leq \mathcal{A}[\![a_2]\!]\,\sigma \\ ff, & \text{falls } \mathcal{A}[\![a_1]\!]\,\sigma > \mathcal{A}[\![a_2]\!]\,\sigma \end{cases}$$

$$\mathcal{B}[\![\texttt{not } b]\!]\,\sigma = \neg\, \mathcal{B}[\![b]\!]\,\sigma$$
$$\mathcal{B}[\![b_1 \texttt{ and } b_2]\!]\,\sigma = \mathcal{B}[\![b_1]\!]\,\sigma \wedge \mathcal{B}[\![b_2]\!]\,\sigma$$

Transitionssysteme

$(\Gamma, T, \rhd)$

- Menge der Konfigurationen Γ
- Menge der terminalen Konfigurationen $T \subseteq \Gamma$
- Transitionsrelation $\rhd \subseteq (\Gamma \setminus T) \times \Gamma$

- Deterministisches Transitionssystem

  falls $\gamma \rhd \gamma'$ und $\gamma \rhd \gamma''$, dann $\gamma' = \gamma''$
- Festgefahrene Konfiguration $\gamma \in \Gamma \setminus T$

  es gibt kein γ', sodaß $\gamma \rhd \gamma'$

IMP: Natürliche Semantik von Anweisungen

Konfigurationen $\langle S, \sigma \rangle \in \mathrm{Stm} \times \Sigma, \quad \sigma \in \Sigma$

Terminale Konfigurationen $\sigma \in \Sigma$

Transitionen $\langle S, \sigma \rangle \to \sigma'$

(skip$_{\mathrm{ns}}$) $\langle \mathtt{skip}, \sigma \rangle \to \sigma$

(assign$_{\mathrm{ns}}$) $\langle x := a, \sigma \rangle \to \sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma]$

(seq$_{\mathrm{ns}}$) $$\frac{\langle S_1, \sigma \rangle \to \sigma_1 \qquad \langle S_2, \sigma_1 \rangle \to \sigma_2}{\langle S_1 \ ; \ S_2, \sigma \rangle \to \sigma_2}$$

IMP: Natürliche Semantik von Anweisungen

$$(\text{if}^{\text{tt}}_{\text{ns}}) \qquad \frac{\langle S_1, \sigma\rangle \to \sigma_1}{\langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2, \sigma\rangle \to \sigma_1}, \quad \text{falls } \mathcal{B}[\![b]\!]\,\sigma = tt$$

$$(\text{if}^{\text{ff}}_{\text{ns}}) \qquad \frac{\langle S_2, \sigma\rangle \to \sigma_2}{\langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2, \sigma\rangle \to \sigma_2}, \quad \text{falls } \mathcal{B}[\![b]\!]\,\sigma = ff$$

$$(\text{while}^{\text{tt}}_{\text{ns}}) \qquad \frac{\langle S, \sigma\rangle \to \sigma' \quad \langle \texttt{while}\ b\ \texttt{do}\ S, \sigma'\rangle \to \sigma''}{\langle \texttt{while}\ b\ \texttt{do}\ S, \sigma\rangle \to \sigma''}, \quad \text{falls } \mathcal{B}[\![b]\!]\,\sigma = tt$$

$$(\text{while}^{\text{ff}}_{\text{ns}}) \qquad \langle \texttt{while}\ b\ \texttt{do}\ S, \sigma\rangle \to \sigma, \quad \text{falls } \mathcal{B}[\![b]\!]\,\sigma = ff$$

IMP: Semantische Äquivalenzen

Arithmetische Ausdrücke $a, a' \in \mathrm{AExp}$

$a \sim a'$ gdw. $\forall \sigma \in \Sigma \,.\, \mathcal{A}[\![a]\!]\,\sigma = \mathcal{A}[\![a']\!]\,\sigma$

Boolesche Ausdrücke $b, b' \in \mathrm{BExp}$

$b \sim b'$ gdw. $\forall \sigma \in \Sigma \,.\, \mathcal{B}[\![b]\!]\,\sigma = \mathcal{B}[\![b']\!]\,\sigma$

Anweisungen $S, S' \in \mathrm{Stm}$

$S \sim S'$ gdw. $\forall \sigma, \sigma' \in \Sigma \,.\, \langle S, \sigma\rangle \to \sigma' \Leftrightarrow \langle S', \sigma\rangle \to \sigma'$

Wohlfundiertheit

Definition Eine binäre Relation $\prec$ auf einer Menge A heißt wohlfundiert, wenn es keine unendliche absteigende Kette $a_0 \succ a_1 \succ a_2 \succ \cdots$ mit $a_0, a_1, a_2, \ldots \in A$ gibt.

Definition Sei $\prec$ eine binäre Relation auf einer Menge A und sei $M \subseteq A$. Ein Element $m \in A$ heißt minimal in M, falls $m \in M$ und für alle $m' \in A$ mit $m' \prec m$ gilt $m' \notin M$.

Satz Sei $\prec$ eine binäre Relation auf einer Menge A. Die Relation $\prec$ ist genau dann wohlfundiert, wenn jede Menge $\emptyset \neq M \subseteq A$ ein minimales Element besitzt.

Noethersche Induktion

Satz Sei $\prec$ eine wohlfundierte Relation auf einer Menge A und sei P eine Eigenschaft über A. Dann gilt

$$\forall a \in A \,.\, P(a) \iff \forall a \in A \,.\, (\forall b \prec a \,.\, P(b)) \Rightarrow P(a)$$

Beispiel Mathematische Induktion

$m \prec n$, falls $n = m + 1$

Beispiel Strukturelle Induktion

$s \prec t$, falls s direkter Teilterm von t

IMP: Freie Variablen

$$\text{fvar} : \text{AExp} \cup \text{BExp} \cup \text{Stm} \to \wp(\text{Var})$$

$$\text{fvar}(n) = \emptyset, \quad \text{für } n \in \text{Num}$$
$$\text{fvar}(x) = \{x\}, \quad \text{für } x \in \text{Var}$$
$$\text{fvar}(a_1 \; bop \; a_2) = \text{fvar}(a_1) \cup \text{fvar}(a_2), \quad \text{für } bop \in \{+, -, \star\}$$

$$\text{fvar}(nop) = \emptyset, \quad \text{für } nop \in \{\texttt{true}, \texttt{false}\}$$
$$\text{fvar}(a_1 \; bop \; a_2) = \text{fvar}(a_1) \cup \text{fvar}(a_2), \quad \text{für } bop \in \{=, \texttt{<=}, \texttt{and}\}$$
$$\text{fvar}(uop \; b) = \text{fvar}(b), \quad \text{für } uop \in \{\texttt{not}\}$$

$$\text{fvar}(\texttt{skip}) = \emptyset$$
$$\text{fvar}(x \texttt{ := } a) = \{x\} \cup \text{fvar}(a)$$
$$\text{fvar}(S_1 \texttt{ ; } S_2) = \text{fvar}(S_1) \cup \text{fvar}(S_2)$$
$$\text{fvar}(\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2) = \text{fvar}(b) \cup \text{fvar}(S_1) \cup \text{fvar}(S_2)$$
$$\text{fvar}(\texttt{while } b \texttt{ do } S) = \text{fvar}(b) \cup \text{fvar}(S)$$

Wohlfundierte Rekursion

Vorgängerrelation $\prec \subseteq A \times A$, $a \in A$

$$\bullet\prec(\{a\}) = \{a' \in A \mid a' \prec a\}$$

Funktionsrestriktion $f\restriction A' : A' \to B$ für $f : A \to B$ und $A' \subseteq A$

$$f\restriction A' = \{(a, f(a)) \mid a \in A'\}$$

Satz Sei $\prec$ eine wohlfundierte Relation auf einer Menge A.
Gelte $F(a,h) \in B$ für alle $a \in A$ und alle Funktionen $h : \bullet\prec(\{a\}) \to B$.
Dann gibt es genau eine Funktion $f : A \to B$ mit

$$\forall a \in A\,.\, f(a) = F(a, f\restriction\bullet\prec(\{a\}))$$

Induktive Definitionen

Menge von Regelinstanzen R (X/y), X endlich

- Prämissen X, Konklusion y

Definition Eine Menge Q ist R-abgeschlossen, falls für alle $(X/y) \in R$ gilt: $X \subseteq Q \Rightarrow y \in Q$. Für eine Menge Q setze $\hat{R}(Q) = \{y \mid \exists X \subseteq Q \,.\, (X/y) \in R\}$.

Lemma Eine Menge Q ist genau dann R-abgeschlossen, wenn $\hat{R}(Q) \subseteq Q$.

Satz Setze $A = \bigcup_{n \in \mathbb{N}} \hat{R}^n(\emptyset)$. Dann gilt:

1. A ist R-abgeschlossen.
2. $\hat{R}(A) = A$.
3. A ist die kleinste R-abgeschlossene Menge.

Ableitungsinduktion

Menge von Regelinstanzen R

Induktive Definition von R-Ableitungen D_R

$$(\emptyset/y) \in D_R, \quad \text{falls } (\emptyset/y) \in R$$

$$(\{(D_1/x_1), \ldots, (D_n/x_n)\}/y) \in D_R,$$

falls $(\{x_1, \ldots, x_n\}/y) \in R$
und $(D_1/x_1) \in D_R, \ldots, (D_n/x_n) \in D_R$

- Direkte Teilableitung $d \prec_1 e$, falls $e = (D/y)$ und $d \in D$
- Teilableitung $d \prec e$, falls $d \prec_1^+ e$

$d \Vdash_R y$, falls $d = (D/y) \in D_R$

$\Vdash_R y$, falls $d \Vdash_R y$ für ein $d \in D_R$

IMP: Eigenschaften der natürlichen Semantik

Satz Seien $S \in \mathrm{Stm}$ und $\sigma, \sigma', \sigma'' \in \Sigma$. Gilt $\langle S, \sigma \rangle \to \sigma'$ und $\langle S, \sigma \rangle \to \sigma''$, dann ist $\sigma' = \sigma''$.

Satz Seien $S \in \mathrm{Stm}$ und $\sigma, \sigma' \in \Sigma$. Dann gilt $\langle \texttt{while true do } S, \sigma \rangle \not\to \sigma'$.

Regelinduktion

Regelinstanzenmenge R $\quad I_R = \{x \mid \Vdash_R x\}$

Satz I_R ist R-abgeschlossen; und ist Q eine R-abgeschlossene Menge, dann ist $I_R \subseteq Q$.

Satz Sei P eine Eigenschaft über I_R. Dann gilt $\forall x \in I_R \,.\, P(x)$ genau dann, wenn für alle Regelinstanzen $(X/y) \in R$ mit $X \subseteq I_R$ gilt: $(\forall x \in X \,.\, P(x)) \Rightarrow P(y)$.

Satz Sei Q eine Eigenschaft über $A \subseteq I_R$. Dann gilt $\forall a \in A \,.\, Q(a)$ genau dann, wenn für alle Regelinstanzen $(X/y) \in R$ mit $X \subseteq I_R$ und $y \in A$ gilt: $(\forall x \in X \cap A \,.\, Q(x)) \Rightarrow Q(y)$.

IMP: Regelinduktion für Anweisungen

Für eine Eigenschaft P über $\mathrm{Stm} \times \Sigma \times \Sigma$ gilt
$\forall S \in \mathrm{Stm}, \sigma, \sigma' \in \Sigma \,.\, \langle S, \sigma\rangle \to \sigma' \Rightarrow P(S, \sigma, \sigma')$ genau dann, wenn

$$
\begin{aligned}
&\forall \sigma \in \Sigma \,.\, P(\texttt{skip}, \sigma, \sigma) \\
\wedge\ &\forall x \in \mathrm{Var}, a \in \mathrm{AExp}, \sigma \in \Sigma \,.\, P(x := a, \sigma, \sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma]) \\
\wedge\ &\forall S_1, S_2 \in \mathrm{Stm}, \sigma, \sigma_1, \sigma_2 \in \Sigma \,. \\
&\quad \langle S_1, \sigma\rangle \to \sigma_1 \wedge P(S_1, \sigma, \sigma_1) \wedge \langle S_2, \sigma_1\rangle \to \sigma_2 \wedge P(S_2, \sigma_1, \sigma_2) \Rightarrow P(S_1 \texttt{ ; } S_2, \sigma, \sigma_2) \\
\wedge\ &\forall b \in \mathrm{BExp}, S_1, S_2 \in \mathrm{Stm}, \sigma, \sigma_1 \in \Sigma \,. \\
&\quad \mathcal{B}[\![b]\!]\,\sigma = tt \wedge \langle S_1, \sigma\rangle \to \sigma_1 \wedge P(S_1, \sigma, \sigma_1) \Rightarrow P(\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2, \sigma, \sigma_1) \\
\wedge\ &\forall b \in \mathrm{BExp}, S_1, S_2 \in \mathrm{Stm}, \sigma, \sigma_2 \in \Sigma \,. \\
&\quad \mathcal{B}[\![b]\!]\,\sigma = ff \wedge \langle S_2, \sigma\rangle \to \sigma_2 \wedge P(S_2, \sigma, \sigma_2) \Rightarrow P(\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2, \sigma, \sigma_2) \\
\wedge\ &\forall b \in \mathrm{BExp}, S \in \mathrm{Stm}, \sigma \in \Sigma \,.\, \mathcal{B}[\![b]\!]\,\sigma = ff \Rightarrow P(\texttt{while } b \texttt{ do } S, \sigma, \sigma) \\
\wedge\ &\forall b \in \mathrm{BExp}, S \in \mathrm{Stm}, \sigma, \sigma', \sigma'' \in \Sigma \,. \\
&\quad \mathcal{B}[\![b]\!]\,\sigma = tt \wedge \langle S, \sigma\rangle \to \sigma' \wedge P(S, \sigma, \sigma') \wedge \\
&\quad \langle \texttt{while } b \texttt{ do } S, \sigma'\rangle \to \sigma'' \wedge P(\texttt{while } b \texttt{ do } S, \sigma', \sigma'') \Rightarrow \\
&\qquad P(\texttt{while } b \texttt{ do } S, \sigma, \sigma'')
\end{aligned}
$$

IMP: Freie änderbare Variablen

$$\mathrm{fvar}_L : \mathrm{Stm} \to \wp\mathrm{Var}$$

$$\mathrm{fvar}_L(\texttt{skip}) = \emptyset$$

$$\mathrm{fvar}_L(x \texttt{ := } a) = \{x\}$$

$$\mathrm{fvar}_L(S_1 \texttt{ ; } S_2) = \mathrm{fvar}_L(S_1) \cup \mathrm{fvar}_L(S_2)$$

$$\mathrm{fvar}_L(\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2) = \mathrm{fvar}_L(S_1) \cup \mathrm{fvar}_L(S_2)$$

$$\mathrm{fvar}_L(\texttt{while } b \texttt{ do } S) = \mathrm{fvar}_L(S)$$

IMP: Strukturell-operationale Semantik

Konfigurationen $\langle S, \sigma \rangle \in \mathrm{Stm} \times \Sigma, \quad \sigma \in \Sigma$

Terminale Konfigurationen $\sigma \in \Sigma$

Transitionen $\langle S, \sigma \rangle \Rightarrow \langle S', \sigma' \rangle, \quad \langle S, \sigma \rangle \Rightarrow \sigma'$

($\mathrm{skip}_{\mathrm{sos}}$) $\langle \mathtt{skip}, \sigma \rangle \Rightarrow \sigma$

($\mathrm{assign}_{\mathrm{sos}}$) $\langle x := a, \sigma \rangle \Rightarrow \sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma]$

($\mathrm{seq}^1_{\mathrm{sos}}$) $$\frac{\langle S_1, \sigma \rangle \Rightarrow \langle S_1', \sigma' \rangle}{\langle S_1 \ ; \ S_2, \sigma \rangle \Rightarrow \langle S_1' \ ; \ S_2, \sigma' \rangle}$$

($\mathrm{seq}^2_{\mathrm{sos}}$) $$\frac{\langle S_1, \sigma \rangle \Rightarrow \sigma'}{\langle S_1 \ ; \ S_2, \sigma \rangle \Rightarrow \langle S_2, \sigma' \rangle}$$

IMP: Strukturell-operationale Semantik

$(\text{if}^{\text{tt}}_{\text{sos}})$ $\quad \langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2, \sigma\rangle \Rightarrow \langle S_1, \sigma\rangle, \quad$ falls $\mathcal{B}[\![b]\!]\,\sigma = tt$

$(\text{if}^{\text{ff}}_{\text{sos}})$ $\quad \langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2, \sigma\rangle \Rightarrow \langle S_2, \sigma\rangle, \quad$ falls $\mathcal{B}[\![b]\!]\,\sigma = ff$

$(\text{while}^{\text{tt}}_{\text{sos}})$ $\quad \langle \texttt{while}\ b\ \texttt{do}\ S, \sigma\rangle \Rightarrow \langle S\ \texttt{;}\ \texttt{while}\ b\ \texttt{do}\ S, \sigma\rangle, \quad$ falls $\mathcal{B}[\![b]\!]\,\sigma = tt$

$(\text{while}^{\text{ff}}_{\text{sos}})$ $\quad \langle \texttt{while}\ b\ \texttt{do}\ S, \sigma\rangle \Rightarrow \sigma, \quad$ falls $\mathcal{B}[\![b]\!]\,\sigma = ff$

IMP: Ableitungen in der strukturell-operationalen Semantik

Endliche Ableitung $\gamma_0, \gamma_1, \ldots, \gamma_k$ $\gamma_0 \Rightarrow^* \gamma_k$

- $\gamma_i \Rightarrow \gamma_{i+1}$ für $0 \leq i < k$ und γ_k terminal oder festgefahren

Unendliche Ableitung $\gamma_0, \gamma_1, \ldots$ $\gamma_0\uparrow$

- $\gamma_i \Rightarrow \gamma_{i+1}$ für $i \geq 0$

Semantische Äquivalenz für Anweisungen $S, S' \in \mathrm{Stm}$

- $\langle S, \sigma\rangle \Rightarrow^* \gamma \Leftrightarrow \langle S', \sigma\rangle \Rightarrow^* \gamma$ für alle $\sigma \in \Sigma$ und terminale oder festgefahrene γ;
- $\langle S, \sigma\rangle\uparrow \Leftrightarrow \langle S', \sigma\rangle\uparrow$ für alle $\sigma \in \Sigma$

IMP: Äquivalenz von natürlicher und strukturell-operationaler Semantik

Satz Seien $S \in \mathrm{Stm}$ und $\sigma, \sigma' \in \Sigma$. Dann gilt

$$\langle S, \sigma \rangle \to \sigma' \quad \Longleftrightarrow \quad \langle S, \sigma \rangle \Rightarrow^* \sigma' \ .$$

IMP: Abbruch

$S \in \mathrm{Stm} ::= \ldots \mid \texttt{abort}$

- Natürliche Semantik
  Keine Erweiterung
- Strukturell-operationale Semantik
  Keine Erweiterung

IMP: Nichtdeterminismus

$$S \in \mathrm{Stm} ::= \ldots \mid S_1 \texttt{ or } S_2$$

- Natürliche Semantik

$$(\mathrm{or}^1_{\mathrm{ns}}) \quad \frac{\langle S_1, \sigma\rangle \to \sigma_1}{\langle S_1 \texttt{ or } S_2, \sigma\rangle \to \sigma_1}$$

$$(\mathrm{or}^2_{\mathrm{ns}}) \quad \frac{\langle S_2, \sigma\rangle \to \sigma_2}{\langle S_1 \texttt{ or } S_2, \sigma\rangle \to \sigma_2}$$

- Strukturell-operationale Semantik

$$(\mathrm{or}^1_{\mathrm{sos}}) \quad \langle S_1 \texttt{ or } S_2, \sigma\rangle \Rightarrow \langle S_1, \sigma\rangle$$

$$(\mathrm{or}^2_{\mathrm{sos}}) \quad \langle S_1 \texttt{ or } S_2, \sigma\rangle \Rightarrow \langle S_2, \sigma\rangle$$

IMP: Parallelität

$$S \in \mathrm{Stm} ::= \ldots \mid S_1 \texttt{ par } S_2$$

- Strukturell-operationale Semantik

$$(\mathrm{par}^1_{\mathrm{sos}}) \quad \frac{\langle S_1, \sigma\rangle \Rightarrow \langle S_1', \sigma_1\rangle}{\langle S_1 \texttt{ par } S_2, \sigma\rangle \Rightarrow \langle S_1' \texttt{ par } S_2, \sigma_1\rangle}$$

$$(\mathrm{par}^2_{\mathrm{sos}}) \quad \frac{\langle S_1, \sigma\rangle \Rightarrow \sigma_1}{\langle S_1 \texttt{ par } S_2, \sigma\rangle \Rightarrow \langle S_2, \sigma_1\rangle}$$

$$(\mathrm{par}^3_{\mathrm{sos}}) \quad \frac{\langle S_2, \sigma\rangle \Rightarrow \langle S_2', \sigma_2\rangle}{\langle S_1 \texttt{ par } S_2, \sigma\rangle \Rightarrow \langle S_1 \texttt{ par } S_2', \sigma_2\rangle}$$

$$(\mathrm{par}^4_{\mathrm{sos}}) \quad \frac{\langle S_2, \sigma\rangle \Rightarrow \sigma_2}{\langle S_1 \texttt{ par } S_2, \sigma\rangle \Rightarrow \langle S_1, \sigma_2\rangle}$$

IMP: Blöcke und Variablendeklarationen

$$V \in \text{VarDecl} ::= \texttt{var}\ x\ \texttt{:=}\ a\ \texttt{;}\ V \mid \varepsilon$$
$$S \in \text{Stm} ::= \ldots \mid \texttt{begin}\ V\ S\ \texttt{end}$$

Deklarierte Variablen $\quad \text{dvar} : \text{VarDecl} \to \wp\text{Var}$

$$\text{dvar}(\varepsilon) = \emptyset$$
$$\text{dvar}(\texttt{var}\ x\ \texttt{:=}\ a\ \texttt{;}\ V) = \{x\} \cup \text{dvar}(V)$$

Variablendeklarationsupdate $\quad upd_V : \text{VarDecl} \times \Sigma \to \Sigma$

$$upd_V(\varepsilon, \sigma) = \sigma$$
$$upd_V(\texttt{var}\ x\ \texttt{:=}\ a\ \texttt{;}\ V, \sigma) = upd_V(V, \sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma])$$

Zustandsrücksetzung $\quad \sigma'[X \mapsto \sigma](x) = \begin{cases} \sigma(x), & \text{falls } x \in X \\ \sigma'(x), & \text{sonst} \end{cases}$

$$(\text{block}_{\text{ns}}) \quad \frac{\langle S, upd_V(V, \sigma)\rangle \to \sigma'}{\langle \texttt{begin}\ V\ S\ \texttt{end}, \sigma\rangle \to \sigma'[\text{dvar}(V) \mapsto \sigma]}$$

IMP: Freie Variablen

$$\mathrm{fvar} : \mathrm{AExp} \cup \mathrm{BExp} \cup \mathrm{Stm} \cup (\mathrm{VarDecl}\ \mathrm{Stm}) \to \wp(\mathrm{Var})$$

$$\mathrm{fvar}(\texttt{begin}\ V\ S\ \texttt{end}) = \mathrm{fvar}(V\ S)$$
$$\mathrm{fvar}(\varepsilon\ S) = \mathrm{fvar}(S)$$
$$\mathrm{fvar}(\texttt{var}\ x\ \texttt{:=}\ a\ \texttt{;}\ V\ S) = \mathrm{fvar}(a) \cup (\mathrm{fvar}(V\ S) \setminus \{x\})$$

$$\mathrm{fvar}_L : \mathrm{Stm} \cup (\mathrm{VarDecl}\ \mathrm{Stm}) \to \wp(\mathrm{Var})$$

$$\mathrm{fvar}_L(\texttt{begin}\ V\ S\ \texttt{end}) = \mathrm{fvar}_L(V\ S)$$
$$\mathrm{fvar}_L(\varepsilon\ S) = \mathrm{fvar}_L(S)$$
$$\mathrm{fvar}_L(\texttt{var}\ x\ \texttt{:=}\ a\ \texttt{;}\ V\ S) = \mathrm{fvar}_L(V\ S) \setminus \{x\}$$

IMP: Variablenumbenennungen

Variablenumbenennung $\delta : \mathrm{Var} \to \mathrm{Var}$

Für $S \in \mathrm{Stm}$, $V \in \mathrm{VarDecl}$ sei $x_{\delta,V\,S}\ \varepsilon\ \mathrm{Var} \setminus \{\delta(x') \mid x' \in (\mathrm{fvar}(V\ S)) \setminus \{x\}\}$.

$$(\texttt{skip})\delta = \texttt{skip}$$
$$(x := a)\delta = \delta\, x := a\delta$$
$$(S_1\ ;\ S_2)\delta = S_1\delta\ ;\ S_2\delta$$
$$(\texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2)\delta = \texttt{if}\ b\delta\ \texttt{then}\ S_1\delta\ \texttt{else}\ S_2\delta$$
$$(\texttt{while}\ b\ \texttt{do}\ S)\delta = \texttt{while}\ b\delta\ \texttt{do}\ S\delta$$
$$(\texttt{begin}\ V\ S\ \texttt{end})\delta = \texttt{begin}\ (V\ S)\delta\ \texttt{end}$$
$$(\varepsilon\ S)\delta = S\delta$$
$$((\texttt{var}\ x := a\ ;\ V)\ S)\delta = (x_{\delta,V\,S} := a\delta)\ ;\ (V\ S)\delta[x \mapsto x_{\delta,V\,S}]$$

(analog für arithmetische und boolesche Ausdrücke)

IMP: Umbenennungslemma

Lemma Seien $S \in \mathrm{Stm}$, $\sigma, \sigma' \in \Sigma$, δ eine Variablenumbenennung und $X \subseteq \mathrm{Var}$ mit $\mathrm{fvar}(S) \subseteq X$ und $\delta(x) \neq \delta(x')$ für $x \neq x' \in X$.

Sei $\sigma(x) = \sigma'(\delta(x))$ für alle $x \in X$. Dann gibt es für alle $\sigma_1 \in \Sigma$ mit $\langle S, \sigma \rangle \to \sigma_1$, ein $\sigma_2 \in \Sigma$ mit $\langle S\delta, \sigma' \rangle \to \sigma_2$, sodaß $\sigma_1(x) = \sigma_2(\delta(x))$ für alle $x \in X$, und umgekehrt.

Korollar Seien $S \in \mathrm{Stm}$, $\sigma \in \Sigma$ und $x' \notin \mathrm{fvar}(S) \setminus \{x\}$. Dann gilt

$$\texttt{begin var } x' \texttt{ := } a \texttt{ ; } S[x \mapsto x'] \texttt{ end} \sim \texttt{begin var } x \texttt{ := } a \texttt{ ; } S \texttt{ end}$$

IMP: Prozeduren mit dynamischer Bindung

$$p \in \text{Proc}$$
$$P \in \text{ProcDecl} ::= \texttt{proc}\ p\ S\ \texttt{;}\ P \mid \varepsilon$$
$$S \in \text{Stm} ::= \ldots \mid \texttt{begin}\ V\ P\ S\ \texttt{end} \mid \texttt{call}\ p$$

Semantische Kategorien

- Prozedurumgebungen $PEnv = \text{Proc} \rightharpoonup \text{Stm}$
- Konfigurationen $\langle S, \pi, \sigma \rangle \in \text{Stm} \times PEnv \times \Sigma, \quad \sigma \in \Sigma$

Prozedurdeklarationsupdate $upd_P : \text{ProcDecl} \times PEnv \to PEnv$

$$upd_P(\varepsilon, \pi) = \pi$$
$$upd_P(\texttt{proc}\ p\ S\ \texttt{;}\ P, \pi) = upd_P(P, \pi[p \mapsto S])$$

$$(\text{block}_{\text{ns}}) \quad \frac{\langle S, upd_P(P,\pi), upd_V(V,\sigma) \rangle \to \sigma'}{\langle \texttt{begin}\ V\ P\ S\ \texttt{end}, \pi, \sigma \rangle \to \sigma'[\text{dvar}(V) \mapsto \sigma]}$$

$$(\text{call}_{\text{ns}}) \quad \frac{\langle S, \pi, \sigma \rangle \to \sigma'}{\langle \texttt{call}\ p, \pi, \sigma \rangle \to \sigma'}, \quad \text{falls } \pi(p) = S$$

IMP: Prozeduren mit statischer Bindung

$$p \in \text{Proc}$$
$$P \in \text{ProcDecl} ::= \texttt{proc}\ p\ S\ \texttt{;}\ P \mid \varepsilon$$
$$S \in \text{Stm} ::= \ldots \mid \texttt{begin}\ V\ P\ S\ \texttt{end} \mid \texttt{call}\ p$$

Semantische Kategorien

- Prozedurumgebungen $PEnv = \text{Proc} \rightharpoonup (\text{Stm} \times PEnv)$
- Konfigurationen $\langle S, \pi, \sigma \rangle \in \text{Stm} \times PEnv \times \Sigma, \quad \sigma \in \Sigma$

Prozedurdeklarationsupdate $upd_P : \text{ProcDecl} \times PEnv \to PEnv$

$$upd_P(\varepsilon, \pi) = \pi$$
$$upd_P(\texttt{proc}\ p\ S\ \texttt{;}\ P, \pi) = upd_P(P, \pi[p \mapsto (S, \pi)])$$

(block$_{ns}$)
$$\frac{\langle S, upd_P(P,\pi), upd_V(V,\sigma)\rangle \to \sigma'}{\langle \texttt{begin}\ V\ P\ S\ \texttt{end}, \pi, \sigma\rangle \to \sigma'[\text{dvar}(V) \mapsto \sigma]}$$

(call$_{ns}$)
$$\frac{\langle S, \pi_p[p \mapsto (S, \pi_p)], \sigma\rangle \to \sigma'}{\langle \texttt{call}\ p, \pi, \sigma\rangle \to \sigma'}, \quad \text{falls } \pi(p) = (S, \pi_p)$$

IMP: Speicher

Semantische Kategorien

- Speicherzellen Loc
- Speicher $Store = Loc \to \mathbb{Z}$
- Variablenumgebungen $VEnv = \mathrm{Var} \to Loc$
- Konfigurationen $\langle S, \upsilon, \varsigma \rangle \in \mathrm{Stm} \times VEnv \times Store, \quad \varsigma \in Store$

Zustandsrekonstruktion $state : VEnv \to Store \to \Sigma, \qquad state\, \upsilon\, \varsigma = \varsigma \circ \eta$

(assign$_{\mathrm{ns}}$) $\langle x := a, \upsilon, \varsigma \rangle \to \varsigma[\upsilon\, x \mapsto \mathcal{A}[\![a]\!](state\, \upsilon\, \varsigma)]$

IMP: Deklarationen mit statischer Bindung

$$
\begin{aligned}
p &\in \text{Proc} \\
V \in \text{VarDecl} &::= \texttt{var}\ x\ \texttt{:=}\ a\ \texttt{;}\ V \mid \varepsilon \\
P \in \text{ProcDecl} &::= \texttt{proc}\ p\ S\ \texttt{;}\ P \mid \varepsilon \\
S \in \text{Stm} &::= \ldots \mid \texttt{begin}\ V\ P\ S\ \texttt{end} \mid \texttt{call}\ p
\end{aligned}
$$

Semantische Kategorien

- $Loc = \mathbb{N}, \qquad new : Loc \to Loc, \quad new\, l = l + 1$
- $Store = (Loc \to \mathbb{Z}) \times (\{next\} \to Loc)$
- $VEnv = \text{Var} \to Loc$
- $PEnv = \text{Proc} \rightharpoonup (Stm \times VEnv \times PEnv)$
- Konfigurationen $\quad \langle S, \upsilon, \pi, \varsigma \rangle \in \text{Stm} \times VEnv \times PEnv \times Store$

Zustandsrekonstruktion $\quad state : VEnv \to Store \to \Sigma, \quad state\, \upsilon\, \varsigma = (\pi_1\, \varsigma) \circ \eta$

IMP: Deklarationen mit statischer Bindung

Prozedurdeklarationsupdate

$$upd_P : \text{ProcDecl} \times VEnv \times PEnv \to PEnv$$

$$upd_P(\varepsilon, \upsilon, \pi) = \pi$$

$$upd_P(\texttt{proc}\ p\ S\ \texttt{;}\ P, \upsilon, \pi) = upd_P(P, \upsilon, \pi[p \mapsto (S, \upsilon, \pi)])$$

Variablendeklarationsupdate

$$upd_V : \text{VarDecl} \times VEnv \times Store \to VEnv \times Store$$

$$upd_V(\varepsilon, \upsilon, \varsigma) = (\upsilon, \varsigma)$$

$$upd_V(\texttt{var}\ x\ \texttt{:=}\ a\ \texttt{;}\ V, \upsilon, \varsigma) =$$

$$upd_V(V, \upsilon[x \mapsto l], \varsigma[l \mapsto \mathcal{A}[\![a]\!](state\ \upsilon\ \varsigma)][next \mapsto new\ l]) \quad \text{mit } l = \varsigma\ next$$

(block$_{ns}$) $$\frac{\langle S, \upsilon', upd_P(P, \upsilon', \pi), \varsigma' \rangle \to \varsigma''}{\langle \texttt{begin}\ V\ P\ S\ \texttt{end}, \upsilon, \pi, \varsigma \rangle \to \varsigma''}, \quad \text{falls } upd_V(\upsilon, \varsigma) = (\upsilon', \varsigma')$$

(call$_{ns}$) $$\frac{\langle S, \upsilon_p, \pi_p[p \mapsto (S, \upsilon_p, \pi_p)], \varsigma \rangle \to \varsigma'}{\langle \texttt{call}\ p, \upsilon, \pi, \varsigma \rangle \to \varsigma'}, \quad \text{falls } \pi(p) = (S, \upsilon_p, \pi_p)$$

IMP: Referenzen

$$r \in \text{RefExp} ::= x \mid \&x \mid \star r$$

$$a \in \text{AExp} ::= n \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \mid r$$
$$b \in \text{BExp} ::= \ldots \mid r_1 = r_2$$

Semantische Kategorien

- Speicherzellen Loc
- Speicher $Store = Loc \to Loc \uplus \mathbb{Z}$
- Variablenumgebungen $VEnv = \text{Var} \to Loc$

Semantische Funktionen

- $\mathcal{R}[\![-]\!] : \text{RefExp} \to VEnv \to Store \to (Loc \cup \mathbb{Z} \cup \{\bot\})$

$$\mathcal{R}[\![x]\!]\, \upsilon\, \varsigma = \varsigma(\upsilon(x))$$
$$\mathcal{R}[\![\&x]\!]\, \upsilon\, \varsigma = \upsilon(x)$$
$$\mathcal{R}[\![\star r]\!]\, \upsilon\, \varsigma = \begin{cases} \varsigma(\mathcal{R}[\![r]\!]\, \upsilon\, \varsigma), & \text{falls } \mathcal{R}[\![r]\!]\, \upsilon\, \varsigma \in Loc \\ \bot, & \text{sonst} \end{cases}$$

IMP: Referenzen

- $\mathcal{A}[\![-]\!] : \mathrm{AExp} \to \mathit{VEnv} \to \mathit{Store} \to (\mathbb{Z} \cup \{\bot\})$

$$\mathcal{A}[\![n]\!]\, \upsilon\, \varsigma = \mathcal{N}[\![n]\!]$$

$$\mathcal{A}[\![r]\!]\, \upsilon\, \varsigma = \mathcal{R}[\![r]\!]\, \upsilon\, \varsigma$$

$$\mathcal{A}[\![a_1\ bop\ a_2]\!]\, \upsilon\, \varsigma = \begin{cases} \mathcal{A}[\![a_1]\!]\, \upsilon\, \varsigma\ [\![bop]\!]\ \mathcal{A}[\![a_2]\!]\, \upsilon\, \varsigma, \\ \qquad \text{falls } \mathcal{A}[\![a_1]\!]\, \upsilon\, \varsigma, \mathcal{A}[\![a_2]\!]\, \upsilon\, \varsigma \in \mathbb{Z} \\ \bot, \quad \text{sonst} \end{cases}$$

- $\mathcal{B}[\![-]\!] : \mathrm{BExp} \to \mathit{VEnv} \to \mathit{Store} \to (\mathbb{B} \cup \{\bot\})$

$$\mathcal{B}[\![r_1 = r_2]\!]\, \upsilon\, \varsigma = \begin{cases} (\mathcal{R}[\![r_1]\!]\, \upsilon\, \varsigma = \mathcal{R}[\![r_2]\!]\, \upsilon\, \varsigma), \\ \qquad \text{falls } \mathcal{R}[\![r_1]\!]\, \upsilon\, \varsigma, \mathcal{R}[\![r_2]\!]\, \upsilon\, \varsigma \in \mathit{Loc} \vee \\ \qquad\qquad \mathcal{R}[\![r_1]\!]\, \upsilon\, \varsigma, \mathcal{R}[\![r_2]\!]\, \upsilon\, \varsigma \in \mathbb{Z} \\ \bot, \text{sonst} \end{cases}$$

Natürliche Semantik $\quad \langle S, \upsilon, \varsigma \rangle \in \mathrm{Stm} \times \mathit{VEnv} \times \mathit{Store}$

(assign$_{\mathrm{ns}}$) $\quad \langle x := a, \upsilon, \varsigma \rangle \to \varsigma[\upsilon\, x \mapsto \mathcal{A}[\![a]\!]\, \upsilon\, \varsigma]$

IMP: Typen für Referenzen

$$\tau \in \mathit{Type} ::= \mathit{int} \mid \mathit{bool} \mid \mathit{void} \mid \&\tau$$

Typumgebungen $\Gamma = \{x_1 : \tau_1, \ldots, x_n : \tau_n\}$ mit $\forall 1 \leq i \neq j \leq n \,.\, x_i \neq x_j$

- Zugriff $\Gamma(x) = \tau_i$, falls $x = x_i$
- Erweiterung $\Gamma, x : \tau = \Gamma \cup \{x : \tau\}$, falls $\forall 1 \leq i \leq n \,.\, x \neq x_i$

$(\mathsf{num}_{\mathsf{typ}})$ $\Gamma \vdash n : \mathit{int}$

$(\mathsf{var}_{\mathsf{typ}})$ $\Gamma, x : \tau \vdash x : \tau$

$(\mathsf{ref}^1_{\mathsf{typ}})$ $\Gamma, x : \tau \vdash \&x : \&\tau$ $\qquad$ $(\mathsf{ref}^2_{\mathsf{typ}})$ $\dfrac{\Gamma \vdash r : \&\tau}{\Gamma \vdash *r : \tau}$

$(\mathsf{arith}_{\mathsf{typ}})$ $\dfrac{\Gamma \vdash a_1 : \mathit{int} \quad \Gamma \vdash a_2 : \mathit{int}}{\Gamma \vdash a_1 \; \mathit{bop} \; a_2 : \mathit{int}}$ mit $\mathit{bop} \in \{+, -, *\}$

IMP: Typen für Referenzen

$(\text{bool}^1_{\text{typ}})$ $\Gamma \vdash nop : bool$ mit $nop \in \{\texttt{true}, \texttt{false}\}$

$(\text{bool}^2_{\text{typ}})$ $$\frac{\Gamma \vdash a_1 : int \quad \Gamma \vdash a_2 : int}{\Gamma \vdash a_1 \; bop \; a_2 : bool}$$ mit $bop \in \{<=, =\}$

$(\text{bool}^3_{\text{typ}})$ $$\frac{\Gamma \vdash b_1 : bool \quad \Gamma \vdash b_2 : bool}{\Gamma \vdash b_1 \; \texttt{and} \; b_2 : bool}$$

$(\text{bool}^4_{\text{typ}})$ $$\frac{\Gamma \vdash b : bool}{\Gamma \vdash \texttt{not} \; b : bool}$$

$(\text{bool}^5_{\text{typ}})$ $$\frac{\Gamma \vdash r_1 : \tau \quad \Gamma \vdash r_2 : \tau}{\Gamma \vdash r_1 = r_2 : bool}$$

IMP: Typen für Referenzen

$$(\text{skip}_{\text{typ}}) \quad \Gamma \vdash \texttt{skip} : \mathit{void}$$

$$(\text{assign}_{\text{typ}}) \quad \frac{\Gamma \vdash x : \tau \quad \Gamma \vdash a : \tau}{\Gamma \vdash x \texttt{ := } a : \mathit{void}}$$

$$(\text{seq}_{\text{typ}}) \quad \frac{\Gamma \vdash S_1 : \mathit{void} \quad \Gamma \vdash S_2 : \mathit{void}}{\Gamma \vdash S_1 \texttt{ ; } S_2 : \mathit{void}}$$

$$(\text{if}_{\text{typ}}) \quad \frac{\Gamma \vdash b : \mathit{bool} \quad \Gamma \vdash S_1 : \mathit{void} \quad \Gamma \vdash S_2 : \mathit{void}}{\Gamma \vdash \texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2 : \mathit{void}}$$

$$(\text{while}_{\text{typ}}) \quad \frac{\Gamma \vdash b : \mathit{bool} \quad \Gamma \vdash S : \mathit{void}}{\Gamma \vdash \texttt{while } b \texttt{ do } S : \mathit{void}}$$

IMP: Subject Reduction für Referenzen

- Typisierung des Speichers

$$(\text{int}_{\text{sto}}) \quad \varsigma \vdash v : int \qquad (\text{ref}_{\text{sto}}) \quad \frac{\varsigma \vdash \varsigma(l) : \tau}{\varsigma \vdash l : \&\tau}$$

- Kompatibilität $\Gamma \rhd (\upsilon, \varsigma)$

  $\varsigma \vdash \upsilon(x) : \&\Gamma(x)$, falls $\Gamma(x)$ definiert

Lemma Gelte $\Gamma \rhd (\upsilon, \varsigma)$.

1. $\Gamma \vdash r : \tau \Rightarrow \varsigma \vdash \mathcal{R}[\![r]\!]\,\upsilon\,\varsigma : \tau$.
2. $\Gamma \vdash a : int \Rightarrow \mathcal{A}[\![a]\!]\,\upsilon\,\varsigma \in \mathbb{Z}$.
3. $\Gamma \vdash b : bool \Rightarrow \mathcal{B}[\![b]\!]\,\upsilon\,\varsigma \in \mathbb{B}$.

Lemma Gelte $\varsigma \vdash l_1 : \&\tau_1 \wedge \varsigma \vdash z : \tau_1$.

$$\forall l_2 \in Loc\,.\,\varsigma \vdash l_2 : \tau_2 \Rightarrow \varsigma[l_1 \mapsto z] \vdash l_2 : \tau_2\ .$$

Satz Gelte $\Gamma \rhd (\upsilon, \varsigma)$.

$$\Gamma \vdash S : void \wedge \langle S, \upsilon, \varsigma\rangle \to \varsigma' \Rightarrow \Gamma \rhd (\upsilon, \varsigma')\ .$$

IMP: Denotationelle Semantik von Anweisungen

Semantische Relation $\mathcal{S}[\![-]\!] : \mathrm{Stm} \to \wp(\Sigma \times \Sigma)$

$$\mathcal{S}[\![\texttt{skip}]\!] = \{(\sigma, \sigma) \mid \sigma \in \Sigma\}$$

$$\mathcal{S}[\![x := a]\!] = \{(\sigma, \sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma]) \mid \sigma \in \Sigma\}$$

$$\mathcal{S}[\![S_1 \texttt{ ; } S_2]\!] = \mathcal{S}[\![S_2]\!] \circ \mathcal{S}[\![S_1]\!]$$

$$\mathcal{S}[\![\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2]\!] =$$
$$\{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!]\,\sigma = tt \wedge (\sigma, \sigma') \in \mathcal{S}[\![S_1]\!]\}$$
$$\cup \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!]\,\sigma = ff \wedge (\sigma, \sigma') \in \mathcal{S}[\![S_2]\!]\}$$

$$\mathcal{S}[\![\texttt{while } b \texttt{ do } S]\!] =$$
$$\mu(\lambda W.\quad \{(\sigma, \sigma') \mid \mathcal{B}[\![b]\!]\,\sigma = tt \wedge (\sigma, \sigma') \in W \circ \mathcal{S}[\![S]\!]\}$$
$$\cup \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!]\,\sigma = ff\})$$

IMP: Äquivalenz von denotationeller und natürlicher Semantik

Satz Für alle Anweisungen $S \in \mathrm{Stm}$ gilt

$$\mathcal{S}[\![S]\!] = \{(\sigma, \sigma') \mid \langle S, \sigma\rangle \to \sigma'\} \; .$$

Korollar Für jedes $S \in \mathrm{Stm}$ ist $\mathcal{S}[\![S]\!]$ eine partielle Funktion $\Sigma \rightharpoonup \Sigma$.

ω-vollständige partielle Ordnungen

Definition Sei $(P, \sqsubseteq)$ eine partielle Ordnung und sei $X \subseteq P$. Ein $p \in P$ heißt obere Schranke von X, falls $q \sqsubseteq p$ für alle $q \in X$. Ein $p \in P$ heißt kleinste obere Schranke (oder Supremum) von X, falls p eine obere Schranke von X ist und für alle oberen Schranken p' von X gilt: $p \sqsubseteq p'$.

Schreibweisen: $\bigsqcup X$ kleinste obere Schranke von X, falls existent

$$\bigsqcup\{p_1, \ldots, p_n\} = p_1 \sqcup \cdots \sqcup p_n$$

Definition Eine partielle Ordnung $(P, \sqsubseteq)$ heißt ω-vollständig, wenn jede aufsteigende ω-Kette $p_0 \sqsubseteq p_1 \sqsubseteq p_2 \sqsubseteq \ldots$ eine kleinste obere Schranke $\bigsqcup\{p_n \mid n \in \mathbb{N}\}$ in P hat.

Präbereiche und Bereiche

Definition Eine ω-vollständige partielle Ordnung heißt **Präbereich**.
Ein Präbereich $(D, \sqsubseteq)$ heißt **Bereich**, wenn es ein Element $\bot_D \in D$ mit $\bot_D \sqsubseteq d$ für alle $d \in D$ gibt.

Hebung eines Präbereichs P zu einem Bereich $P_\bot$ durch Adjunktion eines kleinsten Elements $\bot$

Definition Ein Präbereich $(P, \sqsubseteq)$ heißt **diskret** (geordnet), falls aus $p \sqsubseteq q$ folgt, daß $p = q$.
Ein Bereich $(D, \sqsubseteq)$ heißt **flach**, falls aus $d \sqsubseteq d'$ folgt, daß $d = \bot_D$ oder $d = d'$.

Vertauschbarkeit von Suprema

Lemma Seien P ein Präbereich und $(p_{m,n})_{(m,n)\in\mathbb{N}\times\mathbb{N}}$ Elemente in P mit $p_{m,n} \sqsubseteq_P p_{m',n'}$, falls $m \leq m'$ und $n \leq n'$. Dann existiert das Supremum der Menge $\{p_{m,n} \mid m,n \in \mathbb{N}\}$ und es gilt

$$\begin{aligned}
&\bigsqcup_P \{p_{m,n} \mid m,n \in \mathbb{N}\} = \\
&\quad \bigsqcup_P \{\bigsqcup_P \{p_{m,n} \mid n \in \mathbb{N}\} \mid m \in \mathbb{N}\} = \\
&\quad \bigsqcup_P \{\bigsqcup_P \{p_{m,n} \mid m \in \mathbb{N}\} \mid n \in \mathbb{N}\} = \\
&\quad \bigsqcup_P \{p_{n,n} \mid n \in \mathbb{N}\} \; .
\end{aligned}$$

Monotone und stetige Funktionen auf Präbereichen

Definition Seien P und Q Präbereiche.
Eine Funktion $f : P \to Q$ heißt monoton, falls für alle $p, p' \in P$ mit $p \sqsubseteq_P p'$ gilt, daß $f(p) \sqsubseteq_Q f(p')$ ist.
Eine Funktion $f : P \to Q$ heißt stetig, falls für alle ω-Ketten $p_0 \sqsubseteq_P p_1 \sqsubseteq_P p_2 \sqsubseteq_P \ldots$ in P gilt:

$$f(\bigsqcup_P \{p_n \mid n \in \mathbb{N}\}) = \bigsqcup_Q \{f(p_n) \mid n \in \mathbb{N}\} .$$

Lemma Seien P, Q Präbereiche und $f : P \to Q$ eine monotone Funktion. Dann ist f genau dann stetig, wenn für alle ω-Ketten $p_0 \sqsubseteq_P p_1 \sqsubseteq_P p_2 \sqsubseteq_P \ldots$ in P gilt:

$$f(\bigsqcup_P \{p_n \mid n \in \mathbb{N}\}) \sqsubseteq_Q \bigsqcup_Q \{f(p_n) \mid n \in \mathbb{N}\} .$$

Funktionsbereiche

Definition Seien P, Q Präbereiche. Dann bezeichnet $[P \to Q]$ die partiell geordnete Menge der stetigen Funktionen von P nach Q mit der punktweisen Ordnung:

$$f \sqsubseteq_{[P \to Q]} g \quad \Longleftrightarrow \quad \forall p \in P \,.\, f(p) \sqsubseteq g(p)$$

Lemma Seien P, Q Präbereiche. Dann ist $[P \to Q]$ ein Präbereich und das Supremum einer ω-Kette $f_0 \sqsubseteq_{[P \to Q]} f_1 \sqsubseteq_{[P \to Q]} f_2 \sqsubseteq_{[P \to Q]} \ldots$ in $[P \to Q]$ ist gegeben durch

$$(\bigsqcup_{[P \to Q]} \{f_n \mid n \in \mathbb{N}\})(p) = \bigsqcup_Q \{f_n(p) \mid n \in \mathbb{N}\} \,.$$

Ist Q ein Bereich, so ist $[P \to Q]$ ein Bereich mit dem kleinsten Element $\bot_{[P \to Q]}$ definiert durch $\bot_{[P \to Q]}(p) = \bot_Q$ für alle $p \in P$.

Stetige Funktionen

Sind P und Q Präbereiche und P diskret, so ist $[P \to Q] \cong P \to Q$.

Ist P ein diskreter Präbereich, so ist $[P \to P_\perp] \cong P \rightharpoonup P$.

P, Q, R Präbereiche

1. Jede konstante Funktion $f : P \to Q$ ist stetig.
2. Die Identitätsfunktion $\mathrm{id} : P \to P$ ist stetig.
3. Sind $f : P \to Q$ und $g : Q \to R$ stetig, dann ist $g \circ f : P \to R$ stetig.
4. Ist $f : P \to Q$ eine stetige Funktion, dann ist $- \circ f : [Q \to R] \to [P \to R]$ stetig.
5. Ist $g : Q \to R$ eine stetige Funktion, dann ist $g \circ - : [P \to Q] \to [P \to R]$ stetig.

Fallunterscheidung

Sei P ein Präbereich und D ein Bereich. Sei
$\text{ite} : [P \to \mathbb{B}] \to [P \to D] \to [P \to D] \to (P \to D)$ definiert durch:

$$\text{ite}(b,f,g)(p) = \begin{cases} f(p), & \text{falls } b(p) = tt \\ g(p), & \text{falls } b(p) = ff \end{cases}$$

Dann ist $\text{ite}(b,f,g)$ für alle $b \in [P \to \mathbb{B}], f, g \in [P \to D]$ eine stetige Funktion von P nach D.
Weiter sind für $b \in [P \to \mathbb{B}]$ und $f, g \in [P \to D]$

$$\text{ite}(b,-,g) : [P \to D] \to [P \to D]$$
$$\text{ite}(b,f,-) : [P \to D] \to [P \to D]$$

stetige Funktionen.

Strikte Funktionen

Definition Seien D und D' Bereiche. Eine Funktion $f : D \to D'$ heißt strikt, falls $f(\bot_D) = \bot_{D'}$.

Definition Seien P, Q Präbereiche und $f : P \to Q$. Dann ist die strikte Hebung $f_\bot : P_\bot \to Q_\bot$ definiert durch

$$f_\bot(x) = \begin{cases} \bot_{Q_\bot}, & \text{falls } x = \bot_{P_\bot} \\ f(x), & \text{sonst} \end{cases} .$$

Definition Seien P ein Präbereich, D ein Bereich und $f : P \to D$. Dann ist die strikte Quellhebung $f_{\bot\!\!\bot} : P_\bot \to D$ definiert durch

$$f_{\bot\!\!\bot}(x) = \begin{cases} \bot_D, & \text{falls } x = \bot_{P_\bot} \\ f(x), & \text{sonst} \end{cases} .$$

Fixpunktsatz von Kleene

Satz Seien D ein Bereich und $f : D \to D$ stetig. Dann ist

$$\mathbf{Y}_D f = \bigsqcup_D \{f^n(\bot_D) \mid n \in \mathbb{N}\}$$

der kleinste Fixpunkt von f.

IMP: Denotationelle Semantik von Anweisungen

Semantische Funktion $\mathcal{S}[\![-]\!] : \mathrm{Stm} \to [\Sigma \to \Sigma_\perp]$

$$\mathcal{S}[\![\texttt{skip}]\!] = \mathrm{id}$$
$$\mathcal{S}[\![x := a]\!] = \lambda\sigma\,.\,\sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma]$$
$$\mathcal{S}[\![S_1 \texttt{ ; } S_2]\!] = \mathcal{S}[\![S_2]\!]_{\perp\!\!\perp} \circ \mathcal{S}[\![S_1]\!]$$
$$\mathcal{S}[\![\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2]\!] = \mathrm{ite}(\mathcal{B}[\![b]\!], \mathcal{S}[\![S_1]\!], \mathcal{S}[\![S_2]\!])$$
$$\mathcal{S}[\![\texttt{while } b \texttt{ do } S]\!] = \mathbf{Y}_{[\Sigma\to\Sigma_\perp]}(\lambda f\,.\,\mathrm{ite}(\mathcal{B}[\![b]\!], f_{\perp\!\!\perp} \circ \mathcal{S}[\![S]\!], \mathrm{id}))$$

Zulässige Eigenschaften und Scott-Induktion

Definition Sei P ein Präbereich. Eine Menge $X \subseteq P$ heißt zulässig (in P), falls $\bigsqcup_P Y \in X$ für alle ω-Ketten Y in X.

Satz Seien P und Q Präbereiche und $f, g \in [P \to Q]$.

1. Die Menge $\{p \in P \mid f\,p \sqsubseteq_Q g\,p\}$ ist zulässig in P.
2. Sind X und X' zulässig in P, so auch $X \cap X'$.

Satz Seien D ein Bereich, $f \in [D \to D]$ und X zulässig in D. Gilt $\bot_D \in X$ und $f(d) \in X$, falls $d \in X$, so auch $\mathbf{Y}_D f \in X$.

Vollständige Verbände

Definition Sei $(P, \sqsubseteq)$ eine partielle Ordnung und sei $X \subseteq P$. Ein $p \in P$ heißt untere Schranke von X, falls $p \sqsubseteq q$ für alle $q \in X$. Ein $p \in P$ heißt größte untere Schranke (oder Infimum) von X, falls p eine untere Schranke von X ist und für alle unteren Schranken p' von X gilt: $p' \sqsubseteq p$.

Schreibweisen: $\bigsqcap X$ größte untere Schranke von X, falls existent

$$\bigsqcap\{p_1, \ldots, p_n\} = p_1 \sqcap \cdots \sqcap p_n$$

Definition Eine partielle Ordnung $(L, \sqsubseteq)$ heißt vollständiger Verband, wenn jede Teilmenge von L eine größte untere Schranke besitzt.

Fixpunktsatz von Knaster und Tarski

Definition Sei $(P, \sqsubseteq)$ eine partielle Ordnung und $f : P \to P$. Ein Element $p \in P$ heißt Präfixpunkt von f, falls $f(p) \sqsubseteq p$; ein Element $p \in P$ heißt Postfixpunkt von f, falls $p \sqsubseteq f(p)$.

Satz Seien $(L, \sqsubseteq)$ ein vollständiger Verband und $f : L \to L$ monoton. Setze

$$m = \bigsqcap \{x \in L \mid f(x) \sqsubseteq x\} \ ,$$
$$M = \bigsqcup \{x \in L \mid x \sqsubseteq f(x)\} \ .$$

Dann sind m und M Fixpunkte von f; m ist der kleinste Präfixpunkt von f, M der größte Postfixpunkt von f.

IMP: Variablendeklarationen

$$S \in \mathrm{Stm} ::= \ldots \mid \texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S$$

Denotationelle Semantik

$$\mathcal{S}[\![\texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S]\!]\,\sigma = (\lambda\sigma' \in \Sigma\,.\,\sigma'[x \mapsto \sigma(x)]) \perp\!\!\!\perp (\mathcal{S}[\![S]\!](\sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma]))$$

Freie Variablen

$$\mathrm{fvar}(\texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S) = (\mathrm{fvar}(S) \setminus \{x\}) \cup \mathrm{fvar}(a)$$

Freie änderbare Variablen

$$\mathrm{fvar}_L(\texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S) = \mathrm{fvar}_L(S) \setminus \{x\}$$

IMP: Koinzidenz- und Umbenennungslemmata

Lemma Seien $S \in \mathrm{Stm}$ und $\sigma, \sigma' \in \Sigma$.

1. Gilt $\sigma x = \sigma' x$ für alle $x \in \mathrm{fvar}(S)$, dann ist entweder $\mathcal{S}[\![S]\!]\,\sigma = \bot = \mathcal{S}[\![S]\!]\,\sigma'$ oder es gilt $(\mathcal{S}[\![S]\!]\,\sigma)x = (\mathcal{S}[\![S]\!]\,\sigma')x$ für alle $x \in \mathrm{fvar}(S)$.
2. Ist $\mathcal{S}[\![S]\!]\,\sigma \neq \bot$, dann ist $(\mathcal{S}[\![S]\!]\,\sigma)x = \sigma x$ für alle $x \notin \mathrm{fvar}_L(S)$.

Lemma Seien $S \in \mathrm{Stm}$, $\sigma, \sigma' \in \Sigma$, δ eine Variablenumbenennung und $V \subseteq \mathrm{Var}$ mit $\mathrm{fvar}(S) \subseteq V$ und $\delta(x) \neq \delta(x')$ für $x \neq x' \in V$.

Gilt $\sigma(x) = \sigma'(\delta(x))$ für alle $x \in V$, dann ist entweder $\mathcal{S}[\![S/\delta]\!]\,\sigma' = \bot = \mathcal{S}[\![S]\!]\,\sigma$ oder es gilt $(\mathcal{S}[\![S/\delta]\!]\,\sigma')(\delta(x)) = (\mathcal{S}[\![S]\!]\,\sigma)x$ für alle $x \in V$.

IMP: Abbruch

$$S \in \mathrm{Stm} ::= \ldots \mid \texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S \mid \texttt{abort}$$

Erweiterter semantischer Bereich $\hat{\Sigma} = \Sigma \cup \{\texttt{abort}\} \times \Sigma$

Erweiterte denotationelle Semantik $\mathcal{S}[\![-]\!] : \mathrm{Stm} \to [\Sigma \to \hat{\Sigma}_\perp]$

Funktionserweiterungen für $f : \Sigma \to \hat{\Sigma}_\perp$

$$f_* : \hat{\Sigma}_\perp \to \hat{\Sigma}_\perp \qquad f_*\perp = \perp, \quad f_*\sigma = f\sigma, \quad f_*(\texttt{abort}, \sigma) = (\texttt{abort}, \sigma)$$

$$f_\dagger : \hat{\Sigma}_\perp \to \hat{\Sigma}_\perp \qquad f_\dagger\perp = \perp, \quad f_\dagger\sigma = f\sigma, \quad f_\dagger(\texttt{abort}, \sigma) = (\texttt{abort}, f\sigma)$$

$$\mathcal{S}[\![\texttt{abort}]\!] = \lambda\sigma\,.\,(\texttt{abort}, \sigma)$$

$$\mathcal{S}[\![S_1\ \texttt{;}\ S_2]\!] = \mathcal{S}[\![S_2]\!]_* \circ \mathcal{S}[\![S_1]\!]$$

$$\mathcal{S}[\![\texttt{while}\ b\ \texttt{do}\ S]\!] = \mathbf{Y}_{[\Sigma \to \hat{\Sigma}_\perp]}(\lambda f\,.\,\mathrm{ite}(\mathcal{B}[\![b]\!], f_* \circ \mathcal{S}[\![S]\!], \mathrm{id}))$$

$$\mathcal{S}[\![\texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S]\!] =$$

$$\lambda\sigma\,.\,(\lambda\sigma' \in \Sigma\,.\,\sigma'[x \mapsto \sigma(x)])_\dagger(\mathcal{S}[\![S]\!](\sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma]))$$

IMP: Ausgaben

$$S \in \mathrm{Stm} ::= \ldots \mid \texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S \mid \texttt{abort} \mid \texttt{out}\ a$$

Erweiterter semantischer Bereich

$$\Omega = \mathbb{Z}^* \cup (\mathbb{Z}^* \times \hat{\Sigma}) \cup \mathbb{Z}^\infty$$
$$\omega \sqsubseteq \omega' \iff \exists \omega'' \in \Omega \,.\, \omega \circ \omega'' = \omega'$$

Injektionen

$$\iota_\perp : \{\langle\rangle\} \to \Omega \qquad \iota_\perp \langle\rangle = \langle\rangle = \perp_\Omega$$
$$\iota_{\text{term}} : \Sigma \to \Omega \qquad \iota_{\text{term}} \sigma = \langle \sigma \rangle$$
$$\iota_{\text{abort}} : \Sigma \to \Omega \qquad \iota_{\text{abort}} \sigma = \langle (\texttt{abort}, \sigma) \rangle$$
$$\iota_{\text{out}} : \mathbb{Z} \times \Omega \to \Omega \qquad \iota_{\text{out}}(n, \omega) = \langle n \rangle \circ \omega$$

IMP: Ausgaben

Funktionserweiterung $f : \Sigma \to \Omega$ zu $f_* : \Omega \to \Omega$

$$
\begin{aligned}
f_*\langle n_0, \ldots, n_{k-1}\rangle &= \langle n_0, \ldots, n_{k-1}\rangle \\
f_*\langle n_0, \ldots, n_{k-1}, \sigma\rangle &= \langle n_0, \ldots, n_{k-1}\rangle \circ (f\,\sigma) \\
f_*\langle n_0, \ldots, n_{k-1}, (\texttt{abort}, \sigma)\rangle &= \langle n_0, \ldots, n_{k-1}, (\texttt{abort}, \sigma)\rangle \\
f_*\langle n_0, n_1, \ldots\rangle &= \langle n_0, n_1, \ldots\rangle
\end{aligned}
$$

Funktionserweiterung $f : \Sigma \to \Sigma$ zu $f_\dagger : \Omega \to \Omega$

$$
\begin{aligned}
f_\dagger\,\langle n_0, \ldots, n_{k-1}\rangle &= \langle n_0, \ldots, n_{k-1}\rangle \\
f_\dagger\,\langle n_0, \ldots, n_{k-1}, \sigma\rangle &= \langle n_0, \ldots, n_{k-1}, f\,\sigma\rangle \\
f_\dagger\,\langle n_0, \ldots, n_{k-1}, (\texttt{abort}, \sigma)\rangle &= \langle n_0, \ldots, n_{k-1}, (\texttt{abort}, f\,\sigma)\rangle \\
f_\dagger\,\langle n_0, n_1, \ldots\rangle &= \langle n_0, n_1, \ldots\rangle
\end{aligned}
$$

IMP: Ausgaben

$$S \in \mathrm{Stm} ::= \ldots \mid \texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S \mid \texttt{abort} \mid \texttt{out}\ a$$

Semantische Funktion $\mathcal{S}[\![-]\!] : \mathrm{Stm} \to [\Sigma \to \Omega]$

$$\mathcal{S}[\![\texttt{skip}]\!] = \iota_{\mathrm{term}}$$
$$\mathcal{S}[\![x\ \texttt{:=}\ a]\!] = \lambda\sigma\,.\,\iota_{\mathrm{term}}\sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma]$$
$$\mathcal{S}[\![S_1\ \texttt{;}\ S_2]\!] = \mathcal{S}[\![S_2]\!]_* \circ \mathcal{S}[\![S_1]\!]$$
$$\mathcal{S}[\![\texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2]\!] = \mathrm{ite}(\mathcal{B}[\![b]\!], \mathcal{S}[\![S_1]\!], \mathcal{S}[\![S_2]\!])$$
$$\mathcal{S}[\![\texttt{while}\ b\ \texttt{do}\ S]\!] = \mathbf{Y}_{[\Sigma\to\Omega]}(\lambda f\,.\,\mathrm{ite}(\mathcal{B}[\![b]\!], f_* \circ \mathcal{S}[\![S]\!], \iota_{\mathrm{term}}))$$
$$\mathcal{S}[\![\texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S]\!] =$$
$$\lambda\sigma\,.\,(\lambda\sigma' \in \Sigma\,.\,\sigma'[x \mapsto \sigma(x)])_\dagger(\mathcal{S}[\![S]\!](\sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma]))$$
$$\mathcal{S}[\![\texttt{out}\ a]\!] = \lambda\sigma\,.\,\iota_{\mathrm{out}}(\mathcal{A}[\![a]\!]\,\sigma, \iota_{\mathrm{term}}\sigma)$$

Direktes Produkt

$P_1, \ldots, P_n$ Präbereiche

Direktes Produkt $P_1 \times \cdots \times P_n \qquad (\prod_{1 \leq i \leq n} P_i)$

- Elemente $\{(p_1, \ldots, p_n) \mid \forall 1 \leq i \leq n \,.\, p_i \in P_i\}$
- Ordnung $(p_1, \ldots, p_n) \sqsubseteq_{P_1 \times \cdots \times P_n} (p'_1, \ldots, p'_n)$, falls $p_i \sqsubseteq_{P_i} p'_i$ für alle $1 \leq i \leq n$

Projektionen (stetig)

- $\pi_i : P_1 \times \cdots \times P_n \to P_i, \quad \pi_i\,(p_1, \ldots, p_n) = p_i$

Fortsetzung auf Funktionen $f_1 : Q \to P_1, \ldots, f_n : Q \to P_n$ stetig

- $(f_1, \ldots, f_n) : Q \to P_1 \times \cdots \times P_n, \quad (f_1, \ldots, f_n)\, q = (f_1\, q, \ldots, f_n\, q)$
- Universelle Eigenschaft $\pi_i \circ (f_1, \ldots, f_n) = f_i$

Direktes Produkt

Lemma Seien Q, $P_1, \ldots, P_n$ Präbereiche und $f : Q \to P_1 \times \cdots \times P_n$ eine Funktion. Dann ist f genau dann stetig, wenn für alle $1 \leq i \leq n$ die Funktionen $\pi_i \circ f : Q \to P_i$ stetig sind.

Lemma Seien Q, $P_1, \ldots, P_n$ Präbereiche und $f : P_1 \times \cdots \times P_n \to Q$ eine Funktion. Dann ist f genau dann stetig, wenn f in jedem Argument stetig ist; d. h., wenn für alle $1 \leq i \leq n$ und alle $p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_n$ die Funktionen $P_i \to Q$ mit $p_i \mapsto f(p_1, \ldots, p_{i-1}, p_i, p_{i+1}, \ldots, p_n)$ stetig sind.

Direkte Summe

$P_1, \ldots, P_n$ Präbereiche

Direkte Summe $P_1 + \cdots + P_n \qquad (\sum_{1 \le i \le n} P_i)$

- Elemente $\{(i, p_i) \mid 1 \le i \le n \wedge p_i \in P_i\}$
- Ordnung $(i, p_i) \sqsubseteq_{P_1 + \cdots + P_n} (j, p'_j)$, falls $i = j$ und $p_i \sqsubseteq_{P_i} p'_j$

Injektionen (stetig)

- $\iota_i : P_i \to P_1 + \cdots + P_n, \quad \iota_i\, p_i = (i, p_i)$

Fortsetzung auf Funktionen $f_1 : P_1 \to Q, \ldots, f_n : P_n \to Q$ stetig

- $[f_1, \ldots, f_n] : P_1 + \cdots + P_n \to Q, \quad [f_1, \ldots, f_n](i, p_i) = f_i\, p_i$
- Universelle Eigenschaft $[f_1, \ldots, f_n] \circ \iota_i = f_i$

Funktionenraum

P, Q Präbereiche

Funktionenraum $[P \to Q]$

- Elemente stetige Funktionen von P nach Q
- Ordnung $f \sqsubseteq_{[P \to Q]} g$, falls $\forall p \in P . f\, p \sqsubseteq_Q g\, p$

Funktionsapplikation (stetig)

- $\mathrm{apply} : [P \to Q] \times P \to Q, \quad \mathrm{apply}(f, p) = f\, p$

Fortsetzung auf Funktionen (Currying) $f : R \times P \to Q$ stetig

- $\mathrm{curry}(f) : R \to [P \to Q], \quad \mathrm{curry}(f)\, r\, p = f(r, p)$
- Universelle Eigenschaft $\mathrm{apply}(\mathrm{curry}(f)\, r, p) = f(r, p)$

Hebung

P Präbereich

Hebung $P_\bot$

- Elemente $P \uplus \{\bot\}$
- Ordnung $p \sqsubseteq_{P_\bot} q$, falls $p \sqsubseteq_P q$ oder $p = \bot$

Einbettung (stetig)

- $\lfloor - \rfloor : P \to P_\bot, \quad \lfloor p \rfloor = p$

Fortsetzung auf Funktionen $f : P \to Q$ stetig für Q Bereich

- $f_{\perp\!\!\!\perp} : P_\bot \to Q, \quad f_{\perp\!\!\!\perp}(\bot) = \bot_Q, \; f_{\perp\!\!\!\perp}(p) = f(p)$
- Universelle Eigenschaft $f_{\perp\!\!\!\perp} \lfloor p \rfloor = f\, p$

Isomorphie von Bereichen

Definition Zwei Bereiche D und E heißen isomorph, $D \cong E$, falls es stetige Funktionen $\varphi : D \to E$ und $\psi : E \to D$ gibt, sodaß gilt: $\psi \circ \varphi = \mathrm{id}_D$ und $\varphi \circ \psi = \mathrm{id}_E$.

Striktes Produkt

$D_1, \ldots, D_n$ Bereiche

Striktes Produkt $D_1 \otimes \cdots \otimes D_n \qquad (\bigotimes_{1 \leq i \leq n} D_i)$

- Elemente
  $\{(d_1, \ldots, d_n) \mid \forall 1 \leq i \leq n \,.\, \bot_{D_i} \neq d_i \in D_i\} \cup \{(\bot_{D_1}, \ldots, \bot_{D_n})\}$
- Ordnung $(d_1, \ldots, d_n) \sqsubseteq_{D_1 \otimes \cdots \otimes D_n} (d'_1, \ldots, d'_n)$, falls $d_i \sqsubseteq_{D_i} d'_i$ für alle $1 \leq i \leq n$

Projektionen (stetig, strikt)

- $\pi_i : D_1 \otimes \cdots \otimes D_n \to D_i, \quad \pi_i (d_1, \ldots, d_n) = d_i$

Strikte Summe

$D_1, \ldots, D_n$ Bereiche

Strikte Summe $D_1 \oplus \cdots \oplus D_n \qquad (\bigoplus_{1 \le i \le n} D_i)$

- Elemente $\{(i, d_i) \mid 1 \le i \le n \wedge \bot_{D_i} \neq d_i \in D_i\} \cup \{\bot\}$
- Ordnung $(i, d_i) \sqsubseteq_{D_1 \oplus \cdots \oplus D_n} (j, d'_j)$, falls $i = j$ und $d_i \sqsubseteq_{D_i} d'_j$, $\bot \sqsubseteq_{D_1 \oplus \cdots \oplus D_n} (i, d_i)$

Injektionen (stetig, strikt)

- $\iota_i : D_i \to D_1 \oplus \cdots \oplus D_n, \quad \iota_i \bot_{D_i} = \bot, \iota_i d_i = (i, d_i) \ \ (d_i \neq \bot_{D_i})$

Strikter Funktionenraum

D, E Bereiche

Strikter Funktionenraum $[D \to_\perp E]$

- Elemente stetige, strikte Funktionen von D nach E
- Ordnung $f \sqsubseteq_{[D \to_\perp E]} g$, falls $\forall d \in D \,.\, f\, d \sqsubseteq_E g\, d$

Funktionsapplikation (stetig, strikt)

- $\text{apply} : [D \to_\perp E] \otimes D \to E, \quad \text{apply}(f, d) = f\, d$

IMP: Eingaben

$$S \in \mathrm{Stm} ::= \ldots \mid \texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S \mid \texttt{abort} \mid \texttt{out}\ a \mid \texttt{in}\ x$$

Erweiterter, rekursiver semantischer Bereich

$$\Omega \cong (\hat{\Sigma} + (\mathbb{Z} \times \Omega) + (\mathbb{Z} \to \Omega))_\perp$$

Isomorphismus $\Omega \underset{\psi}{\overset{\varphi}{\rightleftarrows}} (\hat{\Sigma} + (\mathbb{Z} \times \Omega) + (\mathbb{Z} \to \Omega))_\perp$

Injektionen

$$\iota_{\mathrm{norm}} : \Sigma \to \hat{\Sigma} \qquad \iota_{\mathrm{norm}}\sigma = \sigma$$

$$\iota_{\mathrm{abnorm}} : \Sigma \to \hat{\Sigma} \qquad \iota_{\mathrm{abnorm}}\sigma = (\texttt{abort}, \sigma)$$

$$\psi \circ \lfloor - \rfloor \circ \iota_1 \circ \iota_{\mathrm{norm}} = \iota_{\mathrm{term}} : \Sigma \to \Omega$$

$$\psi \circ \lfloor - \rfloor \circ \iota_1 \circ \iota_{\mathrm{abnorm}} = \iota_{\mathrm{abort}} : \Sigma \to \Omega$$

$$\psi \circ \lfloor - \rfloor \circ \iota_2 = \iota_{\mathrm{out}} : (\mathbb{Z} \times \Omega) \to \Omega$$

$$\psi \circ \lfloor - \rfloor \circ \iota_3 = \iota_{\mathrm{in}} : (\mathbb{Z} \to \Omega) \to \Omega$$

IMP: Eingaben

Funktionserweiterung $f : \Sigma \to \Omega$ zu $f_* : \Omega \to \Omega$

$$
\begin{aligned}
f_* \bot_\Omega &= \bot_\Omega \\
f_*(\iota_{\text{term}}\, \sigma) &= f\, \sigma \\
f_*(\iota_{\text{abort}}\, \sigma) &= \iota_{\text{abort}} \sigma \\
f_*(\iota_{\text{out}}(n, \omega)) &= \iota_{\text{out}}(n, f_*\, \omega) \\
f_*(\iota_{\text{in}}\, g) &= \iota_{\text{in}}(\lambda v\, .\, f_*(g\, v))
\end{aligned}
$$

Funktionserweiterung $f : \Sigma \to \Sigma$ zu $f_\dagger : \Omega \to \Omega$

$$
\begin{aligned}
f_\dagger \bot_\Omega &= \bot_\Omega \\
f_\dagger(\iota_{\text{term}}\, \sigma) &= \iota_{\text{term}}(f\, \sigma) \\
f_\dagger(\iota_{\text{abort}}\, \sigma) &= \iota_{\text{abort}}(f\, \sigma) \\
f_\dagger(\iota_{\text{out}}(n, \omega)) &= \iota_{\text{out}}(n, f_\dagger\, \omega) \\
f_\dagger(\iota_{\text{in}}\, g) &= \iota_{\text{in}}(\lambda v\, .\, f_\dagger(g\, v))
\end{aligned}
$$

IMP: Eingaben

$$S \in \mathrm{Stm} ::= \ldots \mid \texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S \mid \texttt{abort} \mid \texttt{out}\ a \mid \texttt{in}\ x$$

Semantische Funktion $\mathcal{S}[\![-]\!] : \mathrm{Stm} \to [\Sigma \to \Omega]$

$$\mathcal{S}[\![\texttt{skip}]\!] = \iota_{\mathrm{term}}$$
$$\mathcal{S}[\![x\ \texttt{:=}\ a]\!] = \lambda\sigma\,.\,\iota_{\mathrm{term}}\sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma]$$
$$\mathcal{S}[\![S_1\ \texttt{;}\ S_2]\!] = \mathcal{S}[\![S_2]\!]_* \circ \mathcal{S}[\![S_1]\!]$$
$$\mathcal{S}[\![\texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2]\!] = \mathrm{ite}(\mathcal{B}[\![b]\!], \mathcal{S}[\![S_1]\!], \mathcal{S}[\![S_2]\!])$$
$$\mathcal{S}[\![\texttt{while}\ b\ \texttt{do}\ S]\!] = \mathbf{Y}_{[\Sigma\to\Omega]}(\lambda f\,.\,\mathrm{ite}(\mathcal{B}[\![b]\!], f_* \circ \mathcal{S}[\![S]\!], \iota_{\mathrm{term}}))$$
$$\mathcal{S}[\![\texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S]\!] =$$
$$\lambda\sigma\,.\,(\lambda\sigma' \in \Sigma\,.\,\sigma'[x \mapsto \sigma(x)])_\dagger(\mathcal{S}[\![S]\!](\sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma]))$$
$$\mathcal{S}[\![\texttt{out}\ a]\!] = \lambda\sigma\,.\,\iota_{\mathrm{out}}(\mathcal{A}[\![a]\!]\,\sigma, \iota_{\mathrm{term}}\sigma)$$
$$\mathcal{S}[\![\texttt{in}\ x]\!] = \lambda\sigma\,.\,\iota_{\mathrm{in}}(\lambda v\,.\,\iota_{\mathrm{term}}(\sigma[x \mapsto v]))$$

IMP: Fortsetzungssemantik

$$S \in \mathrm{Stm} ::= \ldots \mid \texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S$$

Semantische Funktion $\mathcal{C}[\![-]\!] : \mathrm{Stm} \to [\Sigma \to \mathrm{P}] \to [\Sigma \to \mathrm{P}]$

$$\mathcal{C}[\![\texttt{skip}]\!]\,\kappa = \kappa$$
$$\mathcal{C}[\![x\ \texttt{:=}\ a]\!]\,\kappa = \lambda\sigma\,.\,\kappa\,\sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma]$$
$$\mathcal{C}[\![S_1\ \texttt{;}\ S_2]\!]\,\kappa = \mathcal{C}[\![S_1]\!]\,(\mathcal{C}[\![S_2]\!]\,\kappa)$$
$$\mathcal{C}[\![\texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2]\!]\,\kappa = \mathrm{ite}(\mathcal{B}[\![b]\!], \mathcal{C}[\![S_1]\!]\,\kappa, \mathcal{C}[\![S_2]\!]\,\kappa)$$
$$\mathcal{C}[\![\texttt{while}\ b\ \texttt{do}\ S]\!]\,\kappa = \mathbf{Y}_{[\Sigma\to\mathrm{P}]}(\lambda\kappa'\,.\,\mathrm{ite}(\mathcal{B}[\![b]\!], \mathcal{C}[\![S]\!]\,\kappa', \kappa))$$
$$\mathcal{C}[\![\texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S]\!]\,\kappa = \lambda\sigma\,.\,\mathcal{C}[\![S]\!](\lambda\sigma' \in \Sigma\,.\,\kappa\,\sigma'[x \mapsto \sigma(x)])(\sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma])$$

parametrisch in Bereich P

Für $\mathrm{P} = \Sigma_\perp$ $\qquad \mathcal{C}[\![S]\!]\,\kappa\,\sigma = \kappa_{\perp\!\!\perp}(\mathcal{S}[\![S]\!]\,\sigma)$

IMP: Fortsetzungssemantik

$$S \in \mathrm{Stm} ::= \ldots \mid \texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S \mid \texttt{abort} \mid \texttt{out}\ a \mid \texttt{in}\ x$$

$$\mathcal{C}[\![-]\!] : \mathrm{Stm} \to [\Sigma \to \Omega] \to [\Sigma \to \Omega] \to [\Sigma \to \Omega]$$

$$\mathcal{C}[\![\texttt{skip}]\!]\,\kappa_\mathrm{t}\,\kappa_\mathrm{f} = \kappa_\mathrm{t}$$

$$\mathcal{C}[\![x\ \texttt{:=}\ a]\!]\,\kappa_\mathrm{t}\,\kappa_\mathrm{f} = \lambda\sigma\,.\,\kappa_\mathrm{t}\,\sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma]$$

$$\mathcal{C}[\![S_1\ \texttt{;}\ S_2]\!]\,\kappa_\mathrm{t}\,\kappa_\mathrm{f} = \mathcal{C}[\![S_1]\!]\,(\mathcal{C}[\![S_2]\!]\,\kappa_\mathrm{t}\,\kappa_\mathrm{f})\,\kappa_\mathrm{f}$$

$$\mathcal{C}[\![\texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2]\!]\,\kappa_\mathrm{t}\,\kappa_\mathrm{f} = \mathrm{ite}(\mathcal{B}[\![b]\!], \mathcal{C}[\![S_1]\!]\,\kappa_\mathrm{t}\,\kappa_\mathrm{f}, \mathcal{C}[\![S_2]\!]\,\kappa_\mathrm{t}\,\kappa_\mathrm{f})$$

$$\mathcal{C}[\![\texttt{while}\ b\ \texttt{do}\ S]\!]\,\kappa_\mathrm{t}\,\kappa_\mathrm{f} = \mathbf{Y}_{[\Sigma\to\Omega]}(\lambda\kappa'\,.\,\mathrm{ite}(\mathcal{B}[\![b]\!], \mathcal{C}[\![S]\!]\,\kappa'\,\kappa_\mathrm{f}, \kappa_\mathrm{t}))$$

$$\mathcal{C}[\![\texttt{newvar}\ x\ \texttt{:=}\ a\ \texttt{in}\ S]\!]\,\kappa_\mathrm{t}\,\kappa_\mathrm{f} =$$

$$\lambda\sigma\,.\,\mathcal{C}[\![S]\!](\lambda\sigma'\,.\,\kappa_\mathrm{t}\,\sigma'[x \mapsto \sigma(x)])$$

$$(\lambda\sigma'\,.\,\kappa_\mathrm{f}\,\sigma'[x \mapsto \sigma(x)])(\sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma])$$

$$\mathcal{C}[\![\texttt{abort}]\!]\,\kappa_\mathrm{t}\,\kappa_\mathrm{f} = \kappa_\mathrm{f}$$

$$\mathcal{C}[\![\texttt{out}\ a]\!]\,\kappa_\mathrm{t}\,\kappa_\mathrm{f} = \lambda\sigma\,.\,\iota_\mathrm{out}(\mathcal{A}[\![a]\!]\,\sigma, \kappa_\mathrm{t}\,\sigma)$$

$$\mathcal{C}[\![\texttt{in}\ x]\!]\,\kappa_\mathrm{t}\,\kappa_\mathrm{f} = \lambda\sigma\,.\,\iota_\mathrm{in}(\lambda v \in \mathbb{Z}\,.\,\kappa_\mathrm{t}\,\sigma[x \mapsto v])$$

IMP: Ausnahmen

$$e \in \mathrm{Exc}$$

$$S \in \mathrm{Stm} ::= \ldots \mid \texttt{begin}\ S_1\ \texttt{handle}\ e\texttt{:}\ S_2 \mid \texttt{raise}\ e$$

$$\mathcal{C}[\![-]\!] : \mathrm{Stm} \to (\mathrm{Exc} \to [\Sigma \to \mathrm{P}]) \to [\Sigma \to \mathrm{P}] \to [\Sigma \to \mathrm{P}]$$

$$\mathcal{C}[\![\texttt{skip}]\!]\,\eta\,\kappa = \lambda\sigma\,.\,\kappa\sigma$$
$$\mathcal{C}[\![x\ \texttt{:=}\ a]\!]\,\eta\,\kappa = \lambda\sigma\,.\,\kappa(\sigma[x \mapsto \mathcal{A}[\![a]\!]\,\sigma])$$
$$\mathcal{C}[\![S_1\ \texttt{;}\ S_2]\!]\,\eta\,\kappa = \lambda\sigma\,.\,\mathcal{C}[\![S_1]\!]\,\eta(\lambda\sigma'\,.\,\mathcal{C}[\![S_2]\!]\,\eta\,\kappa\,\sigma')\sigma$$
$$\mathcal{C}[\![\texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2]\!]\,\eta\,\kappa = \mathrm{ite}(\mathcal{B}[\![b]\!], \mathcal{C}[\![S_1]\!]\,\eta\,\kappa, \mathcal{C}[\![S_2]\!]\,\eta\,\kappa)$$
$$\mathcal{C}[\![\texttt{while}\ b\ \texttt{do}\ S]\!]\,\eta\,\kappa = \mathbf{Y}_{[\Sigma\to\mathrm{P}]}(\lambda\kappa'\,.\,\mathrm{ite}(\mathcal{B}[\![b]\!], \mathcal{C}[\![S]\!]\,\eta\,\kappa', \kappa))$$
$$\mathcal{C}[\![\texttt{begin}\ S_1\ \texttt{handle}\ e\texttt{:}\ S_2]\!]\,\eta\,\kappa = \mathcal{C}[\![S_1]\!](\eta[e \mapsto \mathcal{C}[\![S_2]\!]\,\eta\,\kappa])\,\kappa$$
$$\mathcal{C}[\![\texttt{raise}\ e]\!]\,\eta\,\kappa = \eta e$$

ASSN: Syntaktische Kategorien

$$
\begin{aligned}
n &\in \mathrm{Num} \\
x &\in \mathrm{Var} \\
X &\in \mathrm{Log}
\end{aligned}
$$

$$
\begin{aligned}
t \in \mathrm{Term} ::= {} & n \mid x \mid X \\
& \mid t_1 + t_2 \mid t_1 - t_2 \mid t_1 \cdot t_2 \\
A \in \mathrm{Frm} ::= {} & tt \mid ff \\
& \mid t_1 = t_2 \mid t_1 \leq t_2 \\
& \mid \neg A \mid A_1 \wedge A_2 \mid A_1 \vee A_2 \mid A_1 \Rightarrow A_2 \\
& \mid \forall X \,.\, A \mid \exists X \,.\, A
\end{aligned}
$$

ASSN: Freie logische Variablen

$$\mathrm{flog} : \mathrm{Term} \cup \mathrm{Frm} \to \wp\mathrm{Log}$$

$$\mathrm{flog}(n) = \mathrm{flog}(x) = \emptyset$$
$$\mathrm{flog}(X) = \{X\}$$
$$\mathrm{flog}(t_1 + t_2) = \mathrm{flog}(t_1 - t_2) = \mathrm{flog}(t_1 \cdot t_2) = \mathrm{flog}(t_1) \cup \mathrm{flog}(t_2)$$

$$\mathrm{flog}(tt) = \mathrm{flog}(ff) = \emptyset$$
$$\mathrm{flog}(t_1 = t_2) = \mathrm{flog}(t_1 \leq t_2) = \mathrm{flog}(t_1) \cup \mathrm{flog}(t_2)$$
$$\mathrm{flog}(\neg A) = \mathrm{flog}(A)$$
$$\mathrm{flog}(A_1 \wedge A_2) = \mathrm{flog}(A_1 \vee A_2) = \mathrm{flog}(A_1 \Rightarrow A_2) = \mathrm{flog}(A_1) \cup \mathrm{flog}(A_2)$$
$$\mathrm{flog}(\forall X \,.\, A) = \mathrm{flog}(\exists X \,.\, A) = \mathrm{flog}(A) \setminus \{X\}$$

ASSN: Substitutionen

Substitution $\zeta : \mathrm{Log} \to \mathrm{Term}$

Für $A \in \mathrm{Frm}$ sei $X_{\zeta,A}\ \varepsilon\ \mathrm{Log} \setminus \bigcup\{\mathrm{flog}(\zeta(X')) \mid X' \in \mathrm{flog}(A) \setminus \{X\}\}$.

$$n\zeta = n, \qquad x\zeta = x, \qquad X\zeta = \zeta X$$

$$(t_1\ bop\ t_2)\zeta = (t_1\zeta)\ bop\ (t_2\zeta), \quad \text{für } bop \in \{+, -, \cdot\}$$

$$(t_1\ bop\ t_2)\zeta = (t_1\zeta)\ bop\ (t_2\zeta), \quad \text{für } bop \in \{=, \leq\}$$

$$(\neg A)\zeta = \neg(A\zeta)$$

$$(A_1\ bop\ A_2)\zeta = (A_1\zeta)\ bop\ (A_2\zeta), \quad \text{für } bop \in \{\wedge, \vee, \Rightarrow\}$$

$$(QX\,.\,A)\zeta = QX_{A,\zeta}\,.\,A(\zeta[X \mapsto X_{A,\zeta}]), \quad \text{für } Q \in \{\forall, \exists\}$$

Analog für Substitutionen $\zeta : \mathrm{Var} \to \mathrm{Term}$

ASSN: Kompositionale Semantik von Termen

Semantischer Bereich

- Interpretationen $\quad Val = \mathrm{Log} \to \mathbb{Z}$

Semantische Funktion $\quad \mathcal{T}[\![-]\!] : \mathrm{Term} \to (Val \to \Sigma \to \mathbb{Z})$

$$\mathcal{T}[\![n]\!]\, I\, \sigma = \mathcal{N}[\![n]\!]$$
$$\mathcal{T}[\![x]\!]\, I\, \sigma = \sigma(x)$$
$$\mathcal{T}[\![X]\!]\, I\, \sigma = I(x)$$

$$\mathcal{T}[\![t_1 + t_2]\!]\, I\, \sigma = \mathcal{T}[\![t_1]\!]\, I\, \sigma + \mathcal{T}[\![t_2]\!]\, I\, \sigma$$
$$\mathcal{T}[\![t_1 - t_2]\!]\, I\, \sigma = \mathcal{T}[\![t_1]\!]\, I\, \sigma - \mathcal{T}[\![t_2]\!]\, I\, \sigma$$
$$\mathcal{T}[\![t_1 \cdot t_2]\!]\, I\, \sigma = \mathcal{T}[\![t_1]\!]\, I\, \sigma \cdot \mathcal{T}[\![t_2]\!]\, I\, \sigma$$

Einbettung von AExp in Term $(+ \mapsto +,\ - \mapsto -,\ \star \mapsto \cdot)$

ASSN: Gültigkeit von Formeln

Semantische Relation $\models \subseteq (\mathit{Val} \times \Sigma) \times \mathrm{Frm}$

$I, \sigma \models tt$

$I, \sigma \models t_1 = t_2$, falls $\mathcal{T}[\![t_1]\!]\, I\, \sigma = \mathcal{T}[\![t_2]\!]\, I\, \sigma$

$I, \sigma \models t_1 \leq t_2$, falls $\mathcal{T}[\![t_1]\!]\, I\, \sigma \leq \mathcal{T}[\![t_2]\!]\, I\, \sigma$

$I, \sigma \models \neg A$, falls $I, \sigma \not\models A$

$I, \sigma \models A_1 \wedge A_2$, falls $I, \sigma \models A_1$ und $I, \sigma \models A_2$

$I, \sigma \models A_1 \vee A_2$, falls $I, \sigma \models A_1$ oder $I, \sigma \models A_2$

$I, \sigma \models A_1 \Rightarrow A_2$, falls $I, \sigma \not\models A_1$ oder $I, \sigma \models A_2$

$I, \sigma \models \forall X\,.\,A$, falls $I[X \mapsto v], \sigma \models A$ für alle $v \in \mathbb{Z}$

$I, \sigma \models \exists X\,.\,A$, falls $I[X \mapsto v], \sigma \models A$ für ein $v \in \mathbb{Z}$

- Erweiterung zu $\models_{\perp\!\!\!\perp} \subseteq (\mathit{Val} \times \Sigma_\perp) \times \mathrm{Frm}$ mit $I, \perp \models_{\perp\!\!\!\perp} A$
- $I \models A \iff \forall \sigma \in \Sigma\,.\, I, \sigma \models A$
- $\models A \iff \forall I \in \mathit{Val},\ \sigma \in \Sigma\,.\, I, \sigma \models A$

Einbettung von BExp in Frm ($\texttt{true} \mapsto tt$, $\texttt{false} \mapsto ff$, $= \mapsto =$, &c.)

ASSN: Substitutionslemmata

Lemma Seien $I \in \mathit{Val}$, $\sigma \in \Sigma$, $t, t' \in \mathrm{Term}$, $A \in \mathrm{Frm}$ und $X \in \mathrm{Log}$.

1. $\mathcal{T}[\![t[X \mapsto t']]\!]\, I\, \sigma = \mathcal{T}[\![t]\!]\, I[X \mapsto \mathcal{T}[\![t']\!]\, I\, \sigma]\, \sigma$;
2. $I, \sigma \models A[X \mapsto t] \iff I[X \mapsto \mathcal{T}[\![t]\!]\, I\, \sigma], \sigma \models A$.

Lemma Seien $I \in \mathit{Val}$, $\sigma \in \Sigma$, $t, t' \in \mathrm{Term}$, $A \in \mathrm{Frm}$ und $x \in \mathrm{Var}$.

1. $\mathcal{T}[\![t[x \mapsto t']]\!]\, I\, \sigma = \mathcal{T}[\![t]\!]\, I\, \sigma[x \mapsto \mathcal{T}[\![t']\!]\, I\, \sigma]$;
2. $I, \sigma \models A[x \mapsto t] \iff I, \sigma[x \mapsto \mathcal{T}[\![t]\!]\, I\, \sigma] \models A$.

IMP/ASSN: Hoare-Tripel für partielle Korrektheit

Partielle Korrektheitsaussage $\{A\}\, S\, \{A'\}$ mit $A, A' \in \mathrm{Frm}$, $S \in \mathrm{Stm}$

Gültigkeitsrelation $I \in \mathit{Val}$, $\sigma \in \Sigma$, $A, A' \in \mathrm{Frm}$

$$I, \sigma \models \{A\}\, S\, \{A'\} \iff (I, \sigma \models A \Rightarrow I, \mathcal{S}[\![S]\!]\, \sigma \models_{\perp\!\!\!\perp} A')$$

- $I \models \{A\}\, S\, \{A'\} \iff \forall \sigma \in \Sigma\,.\, I, \sigma \models \{A\}\, S\, \{A'\}$
- $\models \{A\}\, S\, \{A'\} \iff \forall I \in \mathit{Val},\ \sigma \in \Sigma\,.\, I, \sigma \models \{A\}\, S\, \{A'\}$

Ableitbarkeitsrelation

- $\vdash$ gemäß Hoare-Kalkül für partielle Korrektheit

IMP/ASSN: Hoare-Kalkül für partielle Korrektheit

(skip$_{hp}$) $\{A\}\ \texttt{skip}\ \{A\}$

(assign$_{hp}$) $\{A[x \mapsto a]\}\ x := a\ \{A\}$

(seq$_{hp}$) $$\frac{\{A\}\ S_1\ \{A'\} \qquad \{A'\}\ S_2\ \{A''\}}{\{A\}\ S_1\ ;\ S_2\ \{A''\}}$$

(if$_{hp}$) $$\frac{\{A \wedge b\}\ S_1\ \{A'\} \qquad \{A \wedge \neg b\}\ S_2\ \{A'\}}{\{A\}\ \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2\ \{A'\}}$$

(while$_{hp}$) $$\frac{\{A \wedge b\}\ S\ \{A\}}{\{A\}\ \texttt{while}\ b\ \texttt{do}\ S\ \{A \wedge \neg b\}}$$

(cons$_{hp}$) $$\frac{\{A_1'\}\ S\ \{A_2'\}}{\{A_1\}\ S\ \{A_2\}},$$ falls $\models A_1 \Rightarrow A_1'$, $\models A_2' \Rightarrow A_2$

IMP/ASSN: Korrektheit des Hoare-Kalküls für partielle Korrektheit

Satz Für eine partielle Korrektheitsaussage $\{A\}\, S\, \{A'\}$ gilt

$$\vdash \{A\}\, S\, \{A'\} \quad \Rightarrow \quad \models \{A\}\, S\, \{A'\}\ .$$

IMP/ASSN: Schwächste liberale Vorbedingung

Extension $A \in \mathrm{Frm}$

$$\mathrm{ext}^I(A) = \{\sigma \in \Sigma \mid I, \sigma \models A\}$$

Lemma Seien $I \in \mathit{Val}$, $A, A' \in \mathrm{Frm}$. Dann gilt

$$I \models A \Rightarrow A' \quad \Longleftrightarrow \quad \mathrm{ext}^I(A) \subseteq \mathrm{ext}^I(A') \; .$$

Schwächste liberale Vorbedingung $S \in \mathrm{Stm}$, $A \in \mathrm{Frm}$

$$\mathrm{wlp}^I(S, A) = \{\sigma \in \Sigma \mid I, \mathcal{S}[\![S]\!]\,\sigma \models_{\perp\!\!\!\perp} A\}$$

Lemma Seien $I \in \mathit{Val}$, $A, A' \in \mathrm{Frm}$ und $S \in \mathrm{Stm}$. Dann gilt

$$I \models \{A\}\, S\, \{A'\} \quad \Longleftrightarrow \quad \mathrm{ext}^I(A) \subseteq \mathrm{wlp}^I(S, A') \; .$$

Gödels β-Prädikat

Sei $X \notin \mathrm{flog}(t) \cup \mathrm{flog}(t_0) \cup \mathrm{flog}(t_1)$.

$$\begin{aligned} t = t_0 \bmod t_1 \iff\ & 0 \le t_0 \wedge 0 \le t_1 \wedge \\ & \exists X\,.\, 0 \le X \wedge X \cdot t_1 \le t_0 \wedge \\ & \neg((X+1) \cdot t_1 \le t_0) \wedge t = t_0 - X \cdot t_1 \end{aligned}$$

$$\begin{aligned} F(t_0, t) \iff\ & 0 \le t_0 \wedge \exists X\,.\, 0 \le X \wedge ((t_0 = 2 \cdot X \Rightarrow t = X) \wedge \\ & (t_0 = 2 \cdot X + 1 \Rightarrow t = 0 - X)) \end{aligned}$$

Gödels β-Prädikat für ganze Zahlen $(X' \neq X)$

$$\beta(t_0, t_1, t_2, t_3) \iff \exists X'\,.\,(X' = t_0 \bmod (1 + (1 + t_2) \cdot t_1) \wedge F(X', t_3))$$

Lemma Sei $\langle v_0, \ldots, v_k \rangle \in \mathbb{Z}^+$. Dann existieren $m, n \in \mathbb{N}$, sodaß für alle $0 \le i \le k$ und alle $x \in \mathbb{Z}$ gilt: $\beta(m, n, i, x) \Leftrightarrow x = v_i$.

IMP/ASSN: Ausdrucksstärke

Lemma Seien $S \in \mathrm{Stm}$ und $A \in \mathrm{Frm}$. Dann existiert ein $w(S,A) \in \mathrm{Frm}$, sodaß gilt:

$$\forall I \in \mathit{Val}\,.\ \mathrm{ext}^I(w(S,A)) = \mathrm{wlp}^I(S,A)\ .$$

Lemma Seien $S \in \mathrm{Stm}$, $A \in \mathrm{Frm}$. Sei $w(S,A) \in \mathrm{Frm}$, sodaß $\mathrm{ext}^I(w(S,A)) = \mathrm{wlp}^I(S,A)$ für alle $I \in \mathit{Val}$. Dann gilt:

$$\vdash \{w(S,A)\}\, S\, \{A\}\ .$$

IMP/ASSN: Relative Vollständigkeit des Hoare-Kalküls für partielle Korrektheit

Satz Für eine partielle Korrektheitsaussage $\{A\}\, S\, \{A'\}$ gilt

$$\models \{A\}\, S\, \{A'\} \quad \Rightarrow \quad \vdash \{A\}\, S\, \{A'\}\ .$$

Korollar Die Menge $\{A \in \mathrm{Frm} \mid \models A\}$ ist nicht rekursiv aufzählbar.

IMP/ASSN: Hoare-Tripel für totale Korrektheit

Totale Korrektheitsaussage $[A]\,S\,[A']$ mit $A, A' \in \mathrm{Frm}$, $S \in \mathrm{Stm}$

Gültigkeitsrelation $I \in \mathit{Val}$, $\sigma \in \Sigma$, $A, A' \in \mathrm{Frm}$

$$I, \sigma \models [A]\,S\,[A'] \iff (I, \sigma \models A \Rightarrow \mathcal{S}[\![S]\!]\,\sigma \neq \bot \wedge I, \mathcal{S}[\![S]\!]\,\sigma \models_{\perp\!\!\!\perp} A')$$

- $I \models [A]\,S\,[A'] \iff \forall \sigma \in \Sigma\,.\, I, \sigma \models [A]\,S\,[A']$
- $\models \{A\}\,S\,\{A'\} \iff \forall I \in \mathit{Val},\ \sigma \in \Sigma\,.\, I, \sigma \models [A]\,S\,[A']$

Ableitbarkeitsrelation

- $\vdash$ gemäß Hoare-Kalkül für totale Korrektheit

IMP/ASSN: Hoare-Kalkül für totale Korrektheit

Analog zu partieller Korrektheit

(while$_{\text{ht}}$) $$\frac{[A \wedge b \wedge X = t]\, S\, [A \wedge \neg(X \leq t)]}{[A]\, \texttt{while}\ b\ \texttt{do}\ S\, [A \wedge \neg b]}, \quad \text{falls } \models A \wedge b \Rightarrow 0 \leq t$$

wobei $X \notin \mathrm{flog}(A) \cup \mathrm{flog}(t)$

IMP/ASSN: Bereichstheoretische Charakterisierung

Semantische Funktion $\mathcal{F}[\![-]\!] : \mathrm{Frm} \to (\mathit{Val} \to \Sigma \to \mathbb{B})$

$$\mathcal{F}[\![A]\!]\, I\, \sigma = \begin{cases} \mathit{tt}, & \text{falls } I, \sigma \models A \\ \mathit{ff}, & \text{falls } I, \sigma \not\models A \end{cases}$$

$p_{\mathrm{f}}, p_{\mathrm{t}} : \mathbb{B} \to \{\top\}_\perp$

$$p_{\mathrm{f}}\, x = \begin{cases} \perp, & \text{falls } x = \mathit{tt} \\ \top, & \text{falls } x = \mathit{ff} \end{cases} \qquad p_{\mathrm{t}}\, x = \begin{cases} \top, & \text{falls } x = \mathit{tt} \\ \perp, & \text{falls } x = \mathit{ff} \end{cases}$$

$$I \models \{A\}\, S\, \{A'\} \iff (p_{\mathrm{f}} \circ \mathcal{F}[\![A']\!]\, I)_{\perp\!\!\perp} \circ \mathcal{S}[\![S]\!] \sqsubseteq p_{\mathrm{f}} \circ \mathcal{F}[\![A]\!]\, I$$

$$I \models [A]\, S\, [A'] \iff p_{\mathrm{t}} \circ \mathcal{F}[\![A]\!]\, I \sqsubseteq (p_{\mathrm{t}} \circ \mathcal{F}[\![A']\!]\, I)_{\perp\!\!\perp} \circ \mathcal{S}[\![S]\!]$$

Scott-Topologie

Definition Sei P ein Präbereich. Eine Menge $X \subseteq P$ heißt **Scott-offen** (in P), falls mit $x \in X$ auch $y \in X$ für $x \sqsubseteq_P y$; und falls $\bigsqcup_P Y \in X$ für eine ω-Kette $Y \subseteq P$, so $X \cap Y \neq \emptyset$.

Satz Sei $(P, \sqsubseteq)$ ein Präbereich und sei $\mathcal{O}_P$ die Menge der Scott-offenen Mengen von P. Dann ist $\mathcal{O}_P$ eine Topologie über P.

Definition Die Menge $\mathcal{O}_P$ der Scott-offenen Mengen von P heißt **Scott-Topologie** von P.

Sicherheitseigenschaften

Definition Sei P ein Präbereich. Eine Menge $X \subseteq P$ heißt eine **Sicherheitseigenschaft** (von P), falls mit $x \in X$ auch $y \in X$ für $y \sqsubseteq_P x$; und, falls $Y \subseteq X$ eine ω-Kette ist, auch $\bigsqcup_P Y \in X$ ist.

Satz Sei $(P, \sqsubseteq)$ ein Präbereich. Eine Menge $X \subseteq P$ ist genau dann eine Sicherheitseigenschaft, falls X eine abgeschlossene Menge bzgl. der Scott-Topologie $\mathcal{O}_P$ von P ist.

Algebraische Bereiche

Definition Sei P ein Präbereich. Ein Element $p \in P$ heißt endlich, falls für jede ω-Kette $(p_n)_{n\in\mathbb{N}}$ in P gilt: Ist $p \sqsubseteq_P \bigsqcup_P \{p_n \mid n \in \mathbb{N}\}$, dann gibt es ein $n \in \mathbb{N}$, sodaß $p \sqsubseteq_P p_n$. Die Menge der endlichen Elemente von P wird mit $\mathscr{K}(P)$ bezeichnet.

Definition Ein Präbereich P heißt algebraisch, falls für jedes Element $p \in P$ gilt: Es gibt eine ω-Kette $(e_n)_{n\in\mathbb{N}}$ in $\mathscr{K}(P)$, sodaß $p = \bigsqcup_P \{e_n \mid n \in \mathbb{N}\}$.

IMP/ASSN: Sicherheitseigenschaften

Lemma Seien $A, A' \in \mathrm{Frm}$ und $I \in \mathit{Val}$. Seien $f = p_{\mathrm{f}} \circ \mathcal{F}[\![A']\!]\, I$ und $g = p_{\mathrm{f}} \circ \mathcal{F}[\![A]\!]\, I$. Dann ist

$$Z = \{h \in [\Sigma \to \Sigma_\perp] \mid f_{\perp\!\!\perp} \circ h \sqsubseteq_{[\Sigma \to \Sigma_\perp]} g\}$$

eine Sicherheitseigenschaft von $[\Sigma \to \Sigma_\perp]$.

Z algebraischer Präbereich, falls Σ abzählbar

- endliche Elemente h $\quad |\{\sigma \in \Sigma \mid h\,\sigma \neq \perp\}| < \infty$

Scott-Abschluß

Definition Sei P ein algebraischer Präbereich. Der **Scott-Abschluß** einer Menge $X \subseteq P$ ist gegeben durch die Menge $\overline{X} = \{p \in P \mid \forall e \in \mathscr{K}(P) \,.\, e \sqsubseteq_P p \Rightarrow \exists x \in X \,.\, e \sqsubseteq_P x\}$.

Lemma Sei P ein algebraischer Präbereich und $X \subseteq P$. Dann ist der Scott-Abschluß $\overline{X}$ der Abschluß von X bzgl. der Scott-Topologie $\mathscr{O}_P$ von P.

Lebendigkeitseigenschaften

Definition Sei P ein algebraischer Präbereich. Eine Menge $X \subseteq P$ heißt Lebendigkeitseigenschaft (von P), falls für alle $e \in \mathscr{K}(P)$ ein $x \in X$ existiert mit $e \sqsubseteq_P x$.

Satz Sei P ein algebraischer Präbereich. Eine Menge $X \subseteq P$ ist genau dann eine Lebendigkeitseigenschaft, falls X eine dichte Menge bzgl. der Scott-Topologie $\mathscr{O}_P$ von P ist.

IMP/ASSN: Lebendigkeitseigenschaften

$(A, A') \in \mathrm{Frm} \times \mathrm{Frm}$ total erfüllbar für $I \in \mathit{Val}$

- $\forall \sigma \in \Sigma\,.\ \mathcal{F}[\![A]\!]\, I\, \sigma = \mathit{ff} \quad \vee \quad \exists \sigma' \in \Sigma\,.\ \mathcal{F}[\![A']\!]\, I\, \sigma' \neq \mathit{ff}$

Lemma Seien $A, A' \in \mathrm{Frm}$ und $I \in \mathit{Val}$ und sei (A, A') total erfüllbar für I. Seien $f = p_{\mathrm{f}} \circ \mathcal{F}[\![A']\!]\, I$, $F = p_{\mathrm{t}} \circ \mathcal{F}[\![A]\!]\, I$, $g = p_{\mathrm{f}} \circ \mathcal{F}[\![A]\!]\, I$ und $G = p_{\mathrm{t}} \circ \mathcal{F}[\![A']\!]\, I$. Dann ist

$$L = \{h \in [\Sigma \to \Sigma_\perp] \mid F \sqsubseteq_{[\Sigma \to \Sigma_\perp]} G_{\perp\!\!\perp} \circ h\}$$

eine Lebendigkeitseigenschaft des Präbereichs $Z = \{h \in [\Sigma \to \Sigma_\perp] \mid f_{\perp\!\!\perp} \circ h \sqsubseteq_{[\Sigma \to \Sigma_\perp]} g\}$.