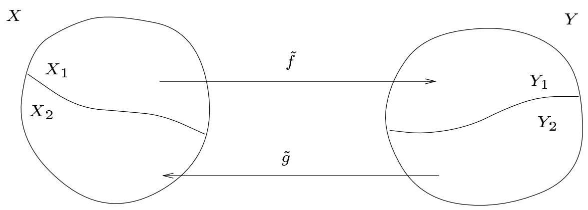
Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Dr. Olaf Müller

WS 1998/99
Lösungsvorschläge zu Blatt 11
8. 2. 1999

Übungen zur Semantik von Programmiersprachen

Aufgabe 1 (Lösungsvorschlag)

(a) Banach'sches Dekompositionstheorem



i)

ii) Wir nehmen an, daß es die gesuchten Partitionierungen von X in X_1 und X_2 bzw. Y in Y_1 und Y_2 gibt. Gesucht ist nun eine Funktion $\Phi : \wp(X) \longrightarrow \wp(X)$, so daß X_2 ein Fixpunkt von Φ ist, d.h. es soll gelten $X_2 = \Phi(X_2)$. Anhand der Zeichnung läßt sich eine solche Funktion konstruieren. Es gilt:

$$\begin{array}{cl} & X_2 \\ = & \{ \text{ Eigenschaft (2)}\} \\ & \tilde{g}(Y_2) \\ = & \{ \text{ Mengentheorie}\} \\ & \tilde{g}(Y \backslash Y_1) \\ = & \{ \text{ Eigenschaft (1)}\} \\ & \tilde{g}(Y \backslash \tilde{f}(X_1)) \\ = & \{ \text{ Mengentheorie}\} \\ & \tilde{g}(Y \setminus \tilde{f}(X \backslash X_2)) \end{array}$$

Wir definieren also Φ folgendermaßen:

$$\Phi(S) = \tilde{g}(Y \setminus \tilde{f}(X \backslash S))$$

für $S \subseteq X$. Für diese Funktion gilt $\Phi(X_2) = X_2$.

iii) Die Frage ist nun, ob ein X_2 existiert, daß diese Fixpunkteigenschaft erfüllt. Dies zeigen wir mit Hilfe des Knaster-Tarski Fixpunkt-Theorems:
$\wp(X)$ ist eine cpo (sogar ein Verband), Φ ist monoton (einfacher Beweis), also existiert ein kleinster Fixpunkt $fix(\Phi)$ mit $\Phi(fix(\Phi)) = fix(\Phi)$. Wir setzen also $X_2 = fix(\Phi)$ und können damit die übrigen Teilmengen berechnen:

$$\begin{array}{rcl} X_1 & = & X \backslash X_1 \\ Y_1 & = & \tilde{f}(X_1) \\ Y_2 & = & Y \backslash Y_1 \end{array}$$

Daß es sich hierbei um Partitionen von X bzw. Y handelt, folgt unmittelbar aus der Definitionen. Die Gültigkeit der Eigenschaft (1) folgt ebenfalls unmittelbar aus der Definition. Jetzt ist noch zu zeigen, daß Eigenschaft (2) gilt:

$$\begin{array}{cc}
 & X_2 \\
= & \{\ X_2 \text{ ist Fixpunkt von } \Phi\ \} \\
 & \Phi(X_2) \\
= & \{\ \text{Def. von } \Phi\ \} \\
 & \tilde{g}(Y \setminus \tilde{f}(X \setminus X_2)) \\
= & \{\ X_1, X_2 \text{ ist Partition von } X\} \\
 & \tilde{g}(Y \setminus \tilde{f}(X_1)) \\
= & \{\ \text{Eigenschaft (1)}\} \\
 & \tilde{g}(Y \setminus Y_1) \\
= & \{\ Y_1, Y_2 \text{ ist Partition von } Y\} \\
 & \tilde{g}(Y_2)
\end{array}$$

(b) Schröder-Bernstein Theorem Aus der Teilaufgabe (a) wissen wir, daß es für die Mengen X und Y Partitionierungen in X_1 und X_2 bzw. Y_1 und Y_2 gibt, sodaß

$$\tilde{f}(X_1) = Y_1 \tag{1}$$
$$\tilde{g}(Y_2) = X_2 \tag{2}$$

Zur Definition von b verwenden wir folgende Hilfsfunktionen, die nur auf Teilmengen von X bzw. Y definiert sind:

$$\hat{f} : X_1 \to Y_1 \ mit \ \hat{f}(x) = f(x)$$

$\hat{f}$ ist injektiv nach Voraussetzung und surjektiv wegen (1), also bijektiv.

$$\hat{g} : Y_2 \to X_2 \ mit \ \hat{g}(x) = g(x)$$

$\hat{g}$ ist injektiv nach Voraussetzung und surjektiv wegen (2), also bijektiv. Damit existiert die Umkehrfunktion $\hat{g}^{-1} : X_2 \to Y_2$.

Wir definieren nun die Funktion $b : X \to Y$ folgendermaßen:

$$b(x) = \begin{cases} \hat{f}(x) & falls\ x \in X_1 \\ \hat{g}^{-1}(x) & falls\ x \in X_2 \end{cases}$$

Daß diese Funktion bijektiv ist, folgt aus der Tatsache, daß sie für jede der Partitionen X_1 und X_2 jeweils durch eine bijektive Funktion definiert ist, deren Wertebereiche Y_1 und Y_2 wieder Partitionen des gesamten Wertebereichs sind.

Aufgabe 2 (Lösungsvorschlag)

Notation: Wie in der Angabe schreiben wir im folgenden d_i für die Komponente i eines Tupels d.

Wir zeigen, daß eine komponentenweise stetige Funktion $f : D^I \to D$ nicht stetig zu sein braucht, durch Angabe einer solchen Funktion. Wir konstruieren diese so systematisch wie möglich.

Dem Hinweis folgend suchen wir eine Funktion $f : \mathbf{O}^\omega \to \mathbf{O}$, wobei $\mathbf{O} = \{\bot, \top\}$ mit $\bot \sqsubseteq \top$ ist. Die Funktion f ist so zu konstruieren, daß gilt:

$$f \text{ ist komponentenweise stetig und} \tag{0}$$

$$\neg(f \text{ ist stetig})\ . \tag{1}$$

Wir sehen uns zuerst (0) näher an.

$$\begin{array}{ll} & f \text{ ist komponentenweise stetig} \\ = & \{\text{Def.}\} \\ & \forall i \in \omega, d \in \mathbf{O}^\omega.(\lambda x \in \mathbf{O}.f(d_1, \ldots, d_{i-1}, x, d_{i+1}, \ldots)) \text{ ist stetig} \end{array}$$

Nun formen wir (1) um.

$$\begin{array}{ll} & \neg(f \text{ ist stetig}) \\ = & \{\ \text{Def.}\} \\ & \neg(f \text{ ist monoton } \wedge (\forall \text{ Ketten } (y.j)_{j\in\omega} \text{ in } \mathbf{O}^\omega.f(\bigsqcup_{j\in\omega} y.j) = \bigsqcup_{j\in\omega}(f(y.j)))) \\ = & \{\text{Prädikatenkalkül}\} \\ & \neg(f \text{ ist monoton}) \vee \\ & (\exists \text{ Kette } (y.j)_{j\in\omega} \text{ in } \mathbf{O}^\omega.f(\bigsqcup_{j\in\omega} y.j) \neq \bigsqcup_{j\in\omega}(f(y.j))) \end{array} \tag{2}$$

Jede komponentenweise stetige (und damit komponentenweise monotone) Funktion ist monoton. Deshalb bleibt für die Konstruktion von f nur die Möglichkeit, (2) zu erfüllen.

Wir suchen also eine komponentenweise stetige Funktion f und eine Kette $(y.j)_{j\in\omega}$ in $\mathbf{O}^\omega$, so daß gilt:

$$f(\bigsqcup_{j\in\omega} y.j) \neq \bigsqcup_{j\in\omega}(f(y.j))$$

. (3)

Aus der komponentenweisen Stetigkeit von f sieht man sofort, daß $(y.j)_{j\in\omega}$ *nicht* nur in endlich vielen Komponenten wachsen kann, da f für solche Ketten $\sqcup$-erhaltend ist. Wir können also $(y.j)_{j\in\omega}$ nur unter den Ketten finden, die in *unendlich* vielen Komponenten wachsen. Eine einfache solche Kette ist die, deren j-tes Kettenglied in allen Komponenten $< j$ den Wert $\top$ und in allen weiteren Komponenten den Wert $\bot$ hat. Formal ausgedrückt definieren wir für alle $j \in \omega$ und $i \in \omega$:

$$(y.j)_i = \begin{cases} \top, & \text{falls } i < j \\ \bot, & \text{falls } i \geq j \end{cases}$$

Dann ist $\bigsqcup_{j\in\omega} y.j$ das Tupel, das in allen Komponenten $\top$ ist.

Nun konstruieren wir eine komponentenweise stetige Funktion f, für die (2) gilt. Da f monoton ist (s.o.), können wir (2) nur erfüllen durch

$$\left.\begin{array}{l} f((\top, \top, \ldots)) = \top \text{ und} \\ f(y.j) = \bot \text{ für alle } j \in \omega\ . \end{array}\right\} (4)$$

Wir definieren nun f durch

$$f(y) = \begin{cases} \bot, & \text{falls } y \text{ nur in endlich vielen Komponenten gleich } \top \text{ ist} \\ \top, & \text{falls } y \text{ in unendlich vielen Komponenten gleich } \top \text{ ist} \end{cases}\ .$$

Dieses f ist komponentenweise stetig und erfüllt (4).

Damit haben wir ein f konstruiert, das komponentenweise stetig, aber nicht stetig ist.

Bemerkung: Das ist nicht die einzige Möglichkeit, f so zu definieren, daß (4) gilt und f komponentenweise stetig ist. Ein anderes f ist

$$f(y) = \begin{cases} \bot, & \text{falls } y \text{ in mindestens einer Komponente gleich } \bot \text{ ist} \\ \top, & \text{falls } y \text{ in allen Komponenten gleich } \top \text{ ist} \end{cases} .$$

Aufgabe 3 (Lösungsvorschlag)

(a) Wir untersuchen das Terminierungsverhalten von $ble(x, y)$ für $x, y \in \mathbf{N}$ unter call-by-value-Auswertung: Wenn $x = 0$ ist, dann terminiert der Aufruf $ble(x, y)$ unmittelbar. Wenn $x \neq 0$ ist, dann terminiert $ble(x, y)$ genau dann, wenn $ble(x - 1, ble(x - y, y))$ terminiert. Ist $x < 0$, so terminiert der äußere Aufruf $ble(x - 1, \ldots)$ nicht. Gilt $x > 0$, so terminiert der Aufruf $ble(x-1, ble(x-y, y))$ genau dann, wenn der innere Aufruf $ble(x-y, y)$ terminiert. Dieser Aufruf terminiert genau dann, wenn sukzessive Subtraktion von y von x irgendwann auf den Wert 0 führt; das ist genau dann der Fall, wenn $y > 0 \wedge y \mid x$ gilt. (Dabei steht "|" für die Teilbarkeitsrelation.)

Insgesamt erhalten wir:

$ble(x, y)$ terminiert für $x, y \in \mathbf{N}$ in call-by-value-Auswertung genau dann, wenn gilt:

$$x = 0 \vee (x > 0 \wedge y > 0 \wedge y \mid x)$$

(b) Da wir hier nur eine zweistellige Funktionsdefinition haben, ist die Funktionsumgebung für call-by-value-Auswertung

$$\mathbf{Fenv} = [\mathbf{N}^2 \to \mathbf{N}_\bot] .$$

Da **Fenv** ein einstelliges Produkt ist, schreiben wir für $\varphi \in \mathbf{Fenv}$ die Projektion φ_1 auf die erste Komponente abkürzend als φ.

Die Variablenumgebung für call-by-value-Auswertung ist

$$\mathbf{Env} = [\mathbf{Var} \to \mathbf{N}] .$$

Im folgenden ist $\rho \in \mathbf{Env}$.

Für $F : \mathbf{Fenv} \to \mathbf{Fenv}$ erhalten wir:

$$\begin{aligned}
& F(\varphi) \\
= \quad & \{\text{Definition von } F \text{ für call-by-value}\} \\
& \lambda k, m \in \mathbf{N} \,.\, [\![\mathbf{if}\ x\ \mathbf{then}\ 1\ \mathbf{else}\ ble(x-1, ble(x-y, y))]\!]_{va}\ \varphi\ \rho[k/x, m/y] \\
= \quad & \{\text{denotationelle call-by-value-Semantik bedingter Terme}\} \\
& \lambda k, m \in \mathbf{N} \,.\, Cond([\![x]\!]_{va}\ \varphi\ \rho[k/x, m/y], [\![1]\!]_{va}\ \varphi\ \rho[k/x, m/y], \\
& [\![ble(x-1, ble(x-y, y))]\!]_{va}\ \varphi\ \rho[k/x, m/y]) \\
= \quad & \{\text{denotationelle call-by-value-Semantik von Variablen, Konstanten}\} \\
& \{\text{und Funktionsapplikationen}\} \\
& \lambda k, m \in \mathbf{N} \,.\, Cond(\lfloor k \rfloor, \lfloor 1 \rfloor,
\end{aligned}$$

$$
\begin{array}{ll}
& \mathit{let}\ v_1 \Leftarrow [\![x-1]\!]_{va}\ \varphi\ \rho[k/x,m/y], v_2 \Leftarrow [\![\mathit{ble}(x-y,y)]\!]_{va}\ \varphi\ \rho[k/x,m/y]\ .\\
& \varphi(v_1,v_2))\\
= & \{\text{denotationelle call-by-value-Semantik von arithmetischen Termen}\}\\
& \{\text{und Funktionsapplikationen}\}\\
& \lambda k,m\in\mathbf{N}\ .\ \mathit{Cond}(\lfloor k\rfloor,\lfloor 1\rfloor,\\
& \mathit{let}\ v_1 \Leftarrow [\![x]\!]_{va}\ \varphi\ \rho[k/x,m/y] -_\perp [\![1]\!]_{va}\ \varphi\ \rho[k/x,m/y],\\
& v_2 \Leftarrow (\mathit{let}\ w_1 \Leftarrow [\![x-y]\!]_{va}\ \varphi\ \rho[k/x,m/y], w_2 \Leftarrow [\![y]\!]_{va}\ \varphi\ \rho[k/x,m/y]\ .\\
& \varphi(w_1,w_2))\ .\ \varphi(v_1,v_2))\\
= & \{\text{denotationelle call-by-value-Semantik von Variablen, Konstanten}\}\\
& \{\text{und arithmetischen Termen}\}\\
& \lambda k,m\in\mathbf{N}\ .\ \mathit{Cond}(\lfloor k\rfloor,\lfloor 1\rfloor,\\
& \mathit{let}\ v_1 \Leftarrow \lfloor k\rfloor -_\perp \lfloor 1\rfloor,\\
& v_2 \Leftarrow (\mathit{let}\ w_1 \Leftarrow [\![x]\!]_{va}\ \varphi\ \rho[k/x,m/y] -_\perp [\![y]\!]_{va}\ \varphi\ \rho[k/x,m/y], w_2 \Leftarrow \lfloor m\rfloor\ .\\
& \varphi(w_1,w_2))\ .\ \varphi(v_1,v_2))\\
= & \{\text{denotationelle call-by-value-Semantik von Variablen}\}\\
& \lambda k,m\in\mathbf{N}\ .\ \mathit{Cond}(\lfloor k\rfloor,\lfloor 1\rfloor,\\
& \mathit{let}\ v_1 \Leftarrow \lfloor k\rfloor -_\perp \lfloor 1\rfloor,\\
& v_2 \Leftarrow (\mathit{let}\ w_1 \Leftarrow \lfloor k\rfloor -_\perp \lfloor m\rfloor, w_2 \Leftarrow \lfloor m\rfloor\ .\ \varphi(w_1,w_2))\ .\ \varphi(v_1,v_2))\\
= & \{\text{Definition von } \mathit{Cond} \text{ und von } -_\perp \text{ und } \lfloor k\rfloor \neq \perp\}\\
& \lambda k,m\in\mathbf{N}\ .\\
& \begin{cases} \lfloor 1\rfloor, & \text{falls } k=0\\ \mathit{let}\ v_1 \Leftarrow \lfloor k-1\rfloor, v_2 \Leftarrow (\mathit{let}\ w_1 \Leftarrow \lfloor k-m\rfloor, w_2 \Leftarrow \lfloor m\rfloor\ .\ \varphi(w_1,w_2))\ .\\ \qquad \varphi(v_1,v_2)), & \text{falls } k\neq 0\end{cases}\\
= & \{\text{Definition von } \mathit{let}\}\\
& \lambda k,m\in\mathbf{N}\ .\\
& \begin{cases} \lfloor 1\rfloor, & \text{falls } k=0\\ \mathit{let}\ v_2 \Leftarrow \varphi(k-m,m)\ .\ \varphi(k-1,v_2), & \text{falls } k\neq 0\end{cases}
\end{array}
$$

(c) Die Teilaufgaben (a) und (b) lassen vermuten, daß gilt:

$$F^n(\perp) = \lambda k,m\in\mathbf{N}\ .\ \begin{cases} \lfloor 1\rfloor, & \text{falls } (k=0 \vee (k>0 \wedge m>0 \wedge m \mid k)) \wedge k<n\\ \perp & \text{sonst}\end{cases}$$

Beweis: Induktion über n:

$n=0$:

$$
\begin{array}{ll}
& F^0(\perp)\\
= & \{F^0 = \mathit{id}\}\\
& \perp\\
= & \{\text{kleinstes Element in } [\mathbf{N}^2 \to \mathbf{N}_\perp] \text{ ist } \lambda k,m\in\mathbf{N}\ .\ \perp\}\\
& \lambda k,m\in\mathbf{N}\ .\ \perp\\
= & \{\text{Arithmetik}\}\\
& \lambda k,m\in\mathbf{N}\ .\ \begin{cases} \lfloor 1\rfloor, & \text{falls } (k=0 \vee (k>0 \wedge m>0 \wedge m \mid k)) \wedge k<0\\ \perp & \text{sonst}\end{cases}
\end{array}
$$

$n \rightsquigarrow n+1$:

$$
\begin{array}{ll}
& F^{n+1}(\bot) \\
= & \{F^{n+1} = F \circ F^n\} \\
& F(F^n(\bot)) \\
= & \{F \text{ (Teilaufgabe (b))}\} \\
& \lambda k, m \in \mathbf{N} \,. \\
& \begin{cases} \lfloor 1 \rfloor, & \text{falls } k = 0 \\ \mathit{let}\ v_2 \Leftarrow (F^n(\bot))(k-m, m)\ .\ (F^n(\bot))(k-1, v_2), & \text{falls } k \neq 0 \end{cases} \\
= & \{\text{Induktionsvoraussetzung und } \beta\text{-Konversion im } \lambda\text{-Kalkül}\} \\
& \lambda k, m \in \mathbf{N} \,. \\
& \begin{cases} \lfloor 1 \rfloor, & \text{falls } k = 0 \\ \mathit{let}\ v_2 \Leftarrow \begin{cases} \lfloor 1 \rfloor, & \text{falls } (k-m = 0 \vee (k-m > 0 \wedge m > 0 \wedge m \mid (k-m))) \\ & \wedge k-m < n \\ \bot & \text{sonst} \end{cases} \\ \ .\begin{cases} \lfloor 1 \rfloor, & \text{falls } (k-1 = 0 \vee (k-1 > 0 \wedge v_2 > 0 \wedge v_2 \mid k-1)) \wedge k-1 < n \\ \bot & \text{sonst} \end{cases} \\ & \text{falls } k \neq 0 \end{cases} \\
= & \{\text{Definition von } \mathit{let}\} \\
& \lambda k, m \in \mathbf{N} \,. \\
& \begin{cases} \lfloor 1 \rfloor, & \text{falls } k = 0 \\ \begin{cases} \lfloor 1 \rfloor, & \text{falls } (k-m = 0 \vee (k-m > 0 \wedge m > 0 \wedge m \mid (k-m))) \wedge \\ & k-m < n \wedge \\ & (k-1 = 0 \vee (k-1 > 0 \wedge 1 > 0 \wedge 1 \mid k-1)) \wedge k-1 < n \wedge \\ & k \neq 0 \\ \bot, & \text{falls } (k-m = 0 \vee (k-m > 0 \wedge m > 0 \wedge m \mid (k-m))) \wedge \\ & k-m < n \wedge \\ & \neg((k-1 = 0 \vee (k-1 > 0 \wedge 1 > 0 \wedge 1 \mid k-1)) \wedge k-1 < n) \wedge \\ & k \neq 0 \end{cases} \\ \bot, & \text{falls } \neg((k-m = 0 \vee (k-m > 0 \wedge m > 0 \wedge m \mid (k-m))) \wedge \\ & k-m < n) \wedge k \neq 0 \end{cases} \\
= & \{\text{längere Umformung der Bedingungen}\} \\
& \lambda k, m \in \mathbf{N} \,. \begin{cases} \lfloor 1 \rfloor, & \text{falls } (k = 0 \vee (k > 0 \wedge m > 0 \wedge m \mid k)) \wedge k < n+1 \\ \bot & \text{sonst} \end{cases}
\end{array}
$$

□

(d) Die explizite Form von $F^n(\bot)$ aus Teilaufgabe (c) läßt vermuten, daß gilt:

$$\bigsqcup_{n \in \omega} F^n(\bot) = \lambda k, m \in \mathbf{N} \,. \begin{cases} \lfloor 1 \rfloor, & \text{falls } k = 0 \vee (k > 0 \wedge m > 0 \wedge m \mid k) \\ \bot & \text{sonst} \end{cases}$$

$\bigsqcup_{n \in \omega} F^n(\bot)$ ist nach Definition die call-by-value Semantik der gegebenen Funktionsdefinition von *ble*.