

Theorie der Programmierung

Wintersemester 2006/07

Übungsblatt 12

Aufgabe 1

Für jeden Typ τ sei $\tau \text{ stream}$ definiert als der Typ $\mu s. \mathbf{unit} \rightarrow \tau * s$. Zeigen Sie: Wenn $\tau_1 <: \tau_2$, dann ist auch $\tau_1 \text{ stream} <: \tau_2 \text{ stream}$.

Aufgabe 2

Vorgegeben seien zwei Typen *vegetable* und *food* mit *vegetable* $<:$ *food*. Darauf aufbauend seien die folgenden Typen definiert:

$$\begin{aligned} \text{cook} &= \langle \text{cooks} : \text{food} \rangle \\ \text{vegetarian_cook} &= \langle \text{cooks} : \text{vegetable} \rangle \\ \text{guest} &= \langle \text{eats} : \text{food} \rightarrow \mathbf{unit} \rangle \\ \text{vegetarian_guest} &= \langle \text{eats} : \text{vegetable} \rightarrow \mathbf{unit} \rangle \end{aligned}$$

- a. Welche Subtyp-Beziehungen gelten zwischen den vier angegebenen Typen?
- b. Wer darf für wen kochen, d.h. welche Typkombinationen τ_1, τ_2 kommen für die Funktion

let *dinner* ($c : \tau_1$) ($g : \tau_2$) = $g \# \text{eats} (c \# \text{cooks})$ **in** ...

in Frage?

Aufgabe 3

Implementieren Sie den Subtyping-Algorithmus für rekursive Typen.