Prof. Dr. J. Giesl
D. Dlugosz

3. Übung zu „Automatisierte Programmverifikation“, SS 03
Abgabe: Mi, 21.05.03, in der Frontalübung

Aufgabe 1 (2 Punkte)

Sei $\to$ eine Relation über einer Menge N und seien $t, q \in N$. Wir schreiben $t \to^n q$ für $n \in \mathbb{N}$, falls die Überführung von t zu q in n Schritten möglich ist (d.h., falls $t \to t_1 \to t_2 \to \ldots \to t_{n-1} \to q$). Dabei bedeutet $t \to^0 q$, dass $t = q$ gilt. Ein Objekt $q \in N$ ist *Normalform* bzgl. der Relation $\to$ gdw. es kein $q' \in N$ mit $q \to q'$ gibt.

Seien t, r, q Terme und sei r ein Teilterm von t. Zeigen Sie, dass falls $t \Rightarrow_P^n q$ und q eine Normalform ist, dann existiert eine Normalform q', so dass $r \Rightarrow_P^m q'$ und $m \leq n$ gilt.

Aufgabe 2 (1.5 + 1.5 + 0.5 Punkte)

Das Programm P bestehe aus den folgenden Algorithmen:

function $\mathsf{plus} : \mathsf{number} \times \mathsf{number} \to \mathsf{number}$
$\mathsf{plus}(\mathcal{O}, y) \equiv y$
$\mathsf{plus}(\mathsf{succ}(x), y) \equiv \mathsf{succ}(\mathsf{plus}(x, y))$

function $\mathsf{null} : \mathsf{number} \times \mathsf{number} \to \mathsf{number}$
$\mathsf{null}(\mathcal{O}, x) \equiv \mathcal{O}$
$\mathsf{null}(\mathsf{succ}(x), y) \equiv \mathsf{null}(x, \mathsf{null}(\mathsf{succ}(x), y))$

function $\mathsf{times} : \mathsf{number} \times \mathsf{number} \to \mathsf{number}$
$\mathsf{times}(\mathcal{O}, y) \equiv \mathcal{O}$
$\mathsf{times}(\mathsf{succ}(x), y) \equiv \mathsf{plus}(y, \mathsf{times}(x, y))$

function $\mathsf{infty} : \mathsf{number} \to \mathsf{number}$
$\mathsf{infty}(x) \equiv \mathsf{infty}(\mathsf{succ}(x))$

a) Berechnen Sie durch Angabe der Auswertungsfolgen die Ergebnisse von
   1) $eval_P(\mathsf{times}(\mathcal{O}, \mathsf{null}(\mathsf{succ}(\mathcal{O}), \mathcal{O})))$,
   2) $eval_P(\mathsf{times}(\mathsf{succ}(\mathsf{plus}(\mathsf{null}(\mathcal{O}, \mathsf{succ}(\mathcal{O})), \mathcal{O})), \mathsf{succ}(\mathcal{O})))$,
   3) $eval_P(\mathsf{null}(\mathsf{times}(\mathsf{plus}(\mathsf{succ}(\mathcal{O}), \mathcal{O}), \mathcal{O}), \mathsf{succ}(\mathsf{infty}(\mathcal{O}))))$.

   Hierbei bezeichnet $eval_P(t)$ die Normalform von t bzgl. der Auswertungsrelation $\Rightarrow_P$.

b) Eine *Outermost*-Auswertungsstrategie ist eine Strategie, bei der jeder Auswertungsschritt soweit außen wie möglich stattfindet. Das bedeutet, dass bei einem Funktionsaufruf $f(t_1, \ldots, t_n)$ nach Möglichkeit anstelle einer Auswertung in den Argumenten $t_1, \ldots, t_n$ immer eine f-Regel ganz außen angewendet wird. (Diese Strategie wird auch *call-by-name* genannt.)

   Bezeichne nun $eval_P^{out}(t)$ die Normalform von t, welche nach einer Outermost-Auswertungsstrategie berechnet wird bzgl. der Auswertungsrelation $\Rightarrow'_P$, für die gilt: $t_1 \Rightarrow'_P t_2$ gdw. $\pi \in Occ(t_1)$, $t_1|_\pi = \sigma(l)$ und $t_2 = t_1[\sigma(r)]_\pi$ für $l \equiv r \in E_P$ und eine Substitution σ. (Im Gegensatz zu der Relation $\Rightarrow_P$ darf hier die Substitution σ Variablen mit beliebigen Termen instantiieren.)
   Berechnen Sie durch Angabe der Auswertungsfolgen die Ergebnisse der Ausdrücke 1), 2) und 3) aus Teilaufgabe a), nachdem in den Ausdrücken $eval_P$ durch $eval_P^{out}$ ersetzt wurde.

c) Können verschiedene Outermost-Auswertungsstrategien unterschiedliches Terminierungsverhalten haben?

Aufgabe 3 (1 + 0.5 Punkte)

Sei $\to_N$ eine Relation über einer Menge N und sei $\to_M$ eine Relation über einer Menge M. Beweisen oder widerlegen Sie die folgenden Aussagen:

a) Sind $\to_N$ und $\to_M$ fundiert, dann ist auch die Relation $\to_K := \to_N \cap \to_M$ über der Menge $N \cap M$ fundiert. Dabei gilt für $t, q \in N \cap M$: $t \to_K q$ gdw. $t \to_N q$ und $t \to_M q$.

b) Sind $\to_N$ und $\to_M$ fundiert, dann ist auch die Relation $\to_K := \to_N \cup \to_M$ über der Menge $N \cup M$ fundiert. Dabei gilt für $t, q \in N \cup M$: $t \to_K q$ gdw. $t \to_N q$ oder $t \to_M q$.