RHEINISCH-
WESTFÄLISCHE
TECHNISCHE
HOCHSCHULE
AACHEN

LEHR- UND FORSCHUNGSGEBIET INFORMATIK II
RWTH Aachen · D-52056 Aachen · GERMANY
http://www-i2.informatik.rwth-aachen.de/lufgi2.html

Prof. Dr. Jürgen Giesl
Darius Dlugosz

Automatisierte Programmverifikation

Übungsblatt 12

Alle Aufgaben werden korrigiert. Die Abgabe ist bis Freitag, 23. Februar 2001, in der Vorlesung oder im Raum 4211 möglich.

Ankündigung 1:

Im nächsten Semester wird vom Lehr- und Forschungsgebiet Informatik II die Vorlesung **Grundlagen der funktionalen Programmierung** *veranstaltet. Hier die Termine:*

V2 Di 10:00 - 11:30 im AH III
V2 Fr 10:00 - 11:30 im AH II
Ü2 Do 15:45 - 17:15 im AH II

Ankündigung 2:

Das Lehr- und Forschungsgebiet Informatik II lädt Sie alle zu einem Informatik-Kolloquium ein, das am 01.03.2001 stattfinden wird.

Thema: Induction and Decision Procedures
Zeit: Donnerstag, 01.03.2001, 17:00 Uhr
Raum: AH I
Referent: Prof. Dr. Deepak Kapur, Department of Computer Science University of New Mexico

Nun zu den Aufgaben:

In folgendem soll die Terminierung von Programmen mit Dependency Pairs gezeigt werden. Führen Sie dabei folgende Schritte aus:

(a) Bestimmen Sie alle Dependency Pairs von P.

(b) Geben Sie für jedes Paar $\langle s, t\rangle \in DP(P)$ die Menge $\mathcal{U}(E_P, t)$ der verwendbaren Gleichungen an.

(c) Bestimmen Sie die Ungleichungen gemäß Satz 6.4.3, die erfüllt sein müssen, damit die Terminierung von P gezeigt werden kann.

(d) Geben Sie das Argument-Filterungssystem $\mathcal{AF}$ an, das Sie verwenden, um die Termienierung zu zeigen.

(e) Geben Sie eine lexikographische Pfadordnung, die die die Ungleichungen aus (c) erfüllt.

Sollten Sie keine lexikographische Pfadordnung finden, versuchen Sie geeignete Relationen $\succ$ und $\succsim$ zu finden, die die Ungleichungen aus (c) erfüllen.

Aufgabe 1: (3 Punkte)

Das Programm P bestehe aus folgenden Datenstrukturen und Algorithmen:

$$
\begin{array}{lcl}
\textbf{structure} \ \mathsf{number} & & \\
\quad \mathcal{O} & : & \mathsf{number} \\
\quad \mathsf{succ} & : & \mathsf{number} \to \mathsf{number} \\
 & & \\
\textbf{function} \ \mathsf{plus} : \mathsf{number} \times \mathsf{number} \to \mathsf{number} & & \\
\quad \mathsf{plus}(\mathcal{O}, y) & \equiv & y \\
\quad \mathsf{plus}(\mathsf{succ}(x), y) & \equiv & \mathsf{succ}(\mathsf{plus}(x, y)) \\
 & & \\
\textbf{function} \ \mathsf{times} : \mathsf{number} \times \mathsf{number} \to \mathsf{number} & & \\
\quad \mathsf{times}(\mathcal{O}, y) & \equiv & \mathcal{O} \\
\quad \mathsf{times}(\mathsf{succ}(x), y) & \equiv & \mathsf{plus}(y, \mathsf{times}(x, y)) \\
 & & \\
\textbf{function} \ \mathsf{pred} : \mathsf{number} \to \mathsf{number} & & \\
\quad \mathsf{pred}(\mathcal{O}) & \equiv & \mathcal{O} \\
\quad \mathsf{pred}(\mathsf{succ}(x)) & \equiv & x \\
 & & \\
\textbf{function} \ \mathsf{minus} : \mathsf{number} \times \mathsf{number} \to \mathsf{number} & & \\
\quad \mathsf{minus}(x, \mathcal{O}) & \equiv & x \\
\quad \mathsf{minus}(x, \mathsf{succ}(y)) & \equiv & \mathsf{minus}(\mathsf{pred}(x), y) \\
 & & \\
\textbf{function} \ \mathsf{ge} : \mathsf{number} \times \mathsf{number} \to \mathsf{bool} & & \\
\quad \mathsf{ge}(x, \mathcal{O}) & \equiv & \mathsf{true} \\
\quad \mathsf{ge}(\mathcal{O}, \mathsf{succ}(y)) & \equiv & \mathsf{false} \\
\quad \mathsf{ge}(\mathsf{succ}(x), \mathsf{succ}(y)) & \equiv & \mathsf{ge}(x, y) \\
 & & \\
\textbf{function} \ \mathsf{if} : \mathsf{bool} \times \mathsf{number} \times \mathsf{number} \to \mathsf{number} & & \\
\quad \mathsf{if}(\mathsf{true}, x, y) & \equiv & x \\
\quad \mathsf{if}(\mathsf{false}, x, y) & \equiv & y \\
 & & \\
\textbf{function} \ \mathsf{gcd} : \mathsf{number} \times \mathsf{number} \to \mathsf{number} & & \\
\quad \mathsf{gcd}(x, \mathcal{O}) & \equiv & x \\
\quad \mathsf{gcd}(\mathcal{O}, \mathsf{succ}(y)) & \equiv & \mathsf{succ}(y) \\
\quad \mathsf{gcd}(\mathsf{succ}(x), \mathsf{succ}(y)) & \equiv & \mathsf{if}(\mathsf{ge}(x, y), \\
 & & \quad \mathsf{gcd}(\mathsf{minus}(x, y), \mathsf{succ}(y)), \\
 & & \quad \mathsf{gcd}(\mathsf{succ}(x), \mathsf{minus}(y, x)))
\end{array}
$$

Versuchen Sie, wie oben beschrieben, die Terminierung von P zu zeigen.

Aufgabe 2: (3 Punkte)

Das Programm P bestehe aus folgenden Datenstrukturen und Algorithmen:

$$
\begin{array}{lcl}
\textbf{structure}\ \ \mathsf{number} & & \\
\quad \mathcal{O} & : & \mathsf{number} \\
\quad \mathsf{succ} & : & \mathsf{number} \to \mathsf{number} \\
 & & \\
\textbf{function}\ \ \mathsf{subtract} : \mathsf{number} \times \mathsf{number} \to \mathsf{number} & & \\
\quad \mathsf{subtract}(x, \mathcal{O}) & \equiv & x \\
\quad \mathsf{subtract}(\mathcal{O}, \mathsf{succ}(y)) & \equiv & \mathcal{O} \\
\quad \mathsf{subtract}(\mathsf{succ}(x), \mathsf{succ}(y)) & \equiv & \mathsf{subtract}(x, y) \\
 & & \\
\textbf{function}\ \ \mathsf{quot} : \mathsf{number} \times \mathsf{number} \to \mathsf{number} & & \\
\quad \mathsf{quot}(x, \mathcal{O}) & \equiv & \mathcal{O} \\
\quad \mathsf{quot}(\mathcal{O}, \mathsf{succ}(y)) & \equiv & \mathcal{O} \\
\quad \mathsf{quot}(\mathsf{succ}(x), \mathsf{succ}(y)) & \equiv & \mathsf{succ}(\mathsf{quot}(\mathsf{subtract}(x, y), \mathsf{succ}(y))) \\
 & & \\
\textbf{function}\ \ \mathsf{log} : \mathsf{number} \to \mathsf{number} & & \\
\quad \mathsf{log}(\mathcal{O}) & \equiv & \mathcal{O} \\
\quad \mathsf{log}(\mathsf{succ}(\mathcal{O})) & \equiv & \mathcal{O} \\
\quad \mathsf{log}(\mathsf{succ}(\mathsf{succ}(x))) & \equiv & \mathsf{succ}(\mathsf{log}(\mathsf{succ}(\mathsf{quot}(x, \mathsf{succ}(\mathsf{succ}(\mathcal{O}))))))
\end{array}
$$

Versuchen Sie, wie oben beschrieben, die Terminierung von P zu zeigen.

Aufgabe 3: (3 Punkte)

Das Programm P bestehe aus folgenden Datenstrukturen und Algorithmen:

$$
\begin{array}{lcl}
\textbf{structure}\ \ \mathsf{number} & & \\
\quad \mathcal{O} & : & \mathsf{number} \\
\quad \mathsf{succ} & : & \mathsf{number} \to \mathsf{number} \\
 & & \\
\textbf{function}\ \ \mathsf{f} : \mathsf{number} \times \mathsf{number} \to \mathsf{number} & & \\
\quad \mathsf{f}(\mathcal{O}, y) & \equiv & \mathsf{succ}(\mathcal{O}) \\
\quad \mathsf{f}(\mathsf{succ}(x), y) & \equiv & \mathsf{f}(\mathsf{f}(x, y), y)
\end{array}
$$

Versuchen Sie, wie oben beschrieben, die Terminierung von P zu zeigen.

Aufgabe 4: (3 Punkte)

Das Programm P bestehe aus folgenden Datenstrukturen und Algorithmen:

$$
\begin{array}{lcl}
\textbf{structure}\ \ \mathsf{number} & & \\
\quad \mathcal{O} & : & \mathsf{number} \\
\quad \mathsf{succ} & : & \mathsf{number} \to \mathsf{number} \\
 & & \\
\multicolumn{3}{l}{\textbf{function}\ \ \mathsf{f} : \mathsf{number} \times \mathsf{number} \to \mathsf{number}} \\
\quad \mathsf{f}(\mathcal{O}, y) & \equiv & \mathcal{O} \\
\quad \mathsf{f}(\mathsf{succ}(x), y) & \equiv & \mathsf{f}(\mathsf{f}(x, y), y)
\end{array}
$$

Versuchen Sie, wie oben beschrieben, die Terminierung von P zu zeigen.

Aufgabe 5: (3 Punkte)

Das Programm P bestehe aus folgenden Datenstrukturen und Algorithmen:

$$
\begin{array}{lcl}
\textbf{structure}\ \ \mathsf{number} & & \\
\quad \mathcal{O} & : & \mathsf{number} \\
\quad \mathsf{succ} & : & \mathsf{number} \to \mathsf{number} \\
 & & \\
\textbf{structure}\ \ \mathsf{list} & & \\
\quad \mathsf{empty} & : & \mathsf{list} \\
\quad \mathsf{add} & : & \mathsf{number} \times \mathsf{list} \to \mathsf{list} \\
 & & \\
\multicolumn{3}{l}{\textbf{function}\ \ \mathsf{append} : \mathsf{list} \times \mathsf{list} \to \mathsf{list}} \\
\quad \mathsf{append}(\mathsf{empty}, l) & \equiv & l \\
\quad \mathsf{append}(\mathsf{add}(n, k), l) & \equiv & \mathsf{add}(n, \mathsf{append}(k, l)) \\
 & & \\
\multicolumn{3}{l}{\textbf{function}\ \ \mathsf{reverse} : \mathsf{list} \to \mathsf{list}} \\
\quad \mathsf{reverse}(\mathsf{empty}) & \equiv & \mathsf{empty} \\
\quad \mathsf{reverse}(\mathsf{add}(n, l)) & \equiv & \mathsf{append}(\mathsf{reverse}(l), \mathsf{add}(n, \mathsf{empty})) \\
 & & \\
\multicolumn{3}{l}{\textbf{function}\ \ \mathsf{shuffle} : \mathsf{list} \to \mathsf{list}} \\
\quad \mathsf{shuffle}(\mathsf{empty}) & \equiv & \mathsf{empty} \\
\quad \mathsf{shuffle}(\mathsf{add}(n, l)) & \equiv & \mathsf{add}(n, \mathsf{shuffle}(\mathsf{reverse}(l)))
\end{array}
$$

Versuchen Sie, wie oben beschrieben, die Terminierung von P zu zeigen.

Aufgabe 6: (2 Punkte)

(a) Sei P ein Programm, $u \in \mathcal{T}(\Sigma, \mathcal{V})$ und σ eine Substitution. Beweisen Sie die folgende Aussage: Wenn σ alle Variablen von u mit Konstruktorgrundtermen instantiiert, dann sind in allen Auswertungen von $\sigma(u)$ nur Regeln aus $\mathcal{U}(E_P, u)$ verwendbar.

(b) Geben Sie ein Beispiel an, in dem $\mathcal{U}(E_P, u)$ zu viele Gleichungen enthält.

Aufgabe 7: (2 Punkte)

Sei P ein Programm, $s \in \mathcal{T}(\Sigma, \mathcal{V})$ und für jede Substitution σ sei $\sigma' = \{x/\sigma(x){\downarrow_{\mathcal{AF}}} \mid x \in DOM(\sigma)\}$. Zeigen Sie: $\sigma(s){\downarrow_{\mathcal{AF}}} = \sigma'(s{\downarrow_{\mathcal{AF}}})$