Übung zur Vorlesung

Automatisierte Logik und Programmierung I

Prof. Chr. Kreitz

Universität Potsdam, Theoretische Informatik — Wintersemester 2016/17

Blatt 5 — Abgabetermin: 12.01.2017

Das fünfte Übungsblatt soll dazu dienen, sich mit der Theorie abhängiger Datentypen und mit der semantischen Typentheorie von Martin-Löf auseinanderzusetzen. Dabei geht es insbesondere um die unterschiedlichen Arten der Erweiterung von Theorien um neue Datentypen.
Wir werden mögliche Lösungen zu Beginn der Veranstaltung am 12.01.2017 besprechen.

Aufgabe 5.1 (Definitorische Erweiterung der Theorie abhängiger Datentypen)

Zeigen Sie, daß in der Typentheorie mit abhängigen Datentypen aus Einheit 7 die folgenden Datentypen als definitorische Erweiterung erklärt werden können

5.1–a Das *abhängige* Produkt $x{:}S{\times}T[x]$ zusammen mit Paarbildung $\langle u\,,v\rangle$ und Analyse von Paaren $\mathsf{match}\ p\ \mathsf{with}\ \langle x\,,y\rangle \mapsto e$

5.1–b Ein leerer Datentyp $\{\}$ ohne Elemente

5.1–c Der Typ $\mathbb{Z}$ der ganzen Zahlen zusammen mit den Basiselementen $\overline{n}$ und $\overline{-n}$, den Operationen $+$ und $*$, sowie einem Operator für primitive Rekursion auf positiven und negativen Zahlen.

Aufgabe 5.2 (Urteile semantisch analysieren)

Zeigen Sie durch Verwendung der formalen Semantikdefinition, daß gilt

5.2–a $\lambda x.\lambda y.(x,y) \in S{\to}T{\to}(S{\times}T)$

5.2–b $\lambda f.\lambda g.\lambda e.\ \mathsf{case}\ e\ \mathsf{of}\ \mathsf{inl}(s)\mapsto f(s)\ \mid\ \mathsf{inr}(t)\mapsto g(t) \in (S{\to}X){\to}(T{\to}X){\to}((S+T){\to}X)$

5.2–c $\lambda f.(\lambda s.f(\mathsf{inl}(s)),\ \lambda t.f(\mathsf{inr}(t))) \in ((S+T){\to}X){\to}((S{\to}X){\times}(T{\to}X))$

Aufgabe 5.3 (Explizite Definition neuer Typen)

Erweitern Sie die Typentheorie aus Einheit 8 durch explizite Definitionen der folgenden Datentypen

5.3–a Den Booleschen Typ $\mathbb{B}$ zusammen mit den Ausdrücken T, F und $\mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ t$.

5.3–b Einen einelementigen Typ `Unit` zusammen mit dem Element `()`.

Aufgabe 5.4 (Konservative Erweiterung der Typentheorie)

5.4–a Zeigen Sie, daß das Produkt $S{\times}T$ zweier Datentypen S und T als konservative Erweiterung definiert werden kann, wenn neben dem abhängigen Funktionenraum $x{:}S{\to}T$ auch der Boolesche Datentyp $\mathbb{B}$ (siehe vorhergehende Aufgabe) zur Verfügung stehen würde.

Beschreiben Sie hierzu $S{\times}T$, $\langle s\,,t\rangle$ und $\mathsf{match}\ p\ \mathsf{with}\ \langle x\,,y\rangle\mapsto t$ als definitorische Abkürzung für entsprechende typentheoretische Ausdrücke und zeigen Sie, daß die so entstehende Semantik der Ausdrücke tatsächlich mit der Semantik aus Einheit 8, Folie 17 übereinstimmt.

5.4–b Zeigen Sie, daß der Boolesche Datentyp $\mathbb{B}$ als konservative Erweiterung definiert werden kann, wenn neben dem abhängigen Funktionenraum $x{:}S{\to}T$ und dem Summentyp $A{+}B$ auch der einelementige Typ `Unit` (siehe vorhergehende Aufgabe) zur Verfügung stehen würde.

Beschreiben Sie hierzu $\mathbb{B}$, T, F und $\mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ t$ als definitorische Abkürzung für entsprechende typentheoretische Ausdrücke und zeigen Sie, daß die so entstehende Semantik der Ausdrücke tatsächlich mit der Semantik von $\mathbb{B}$ übereinstimmt.

Aufgabe 5.5 (Für Tüftler: Lehre der Leere)

Zeigen Sie, das sich der Datentyp Void durch den Ausdruck $\mathsf{T} = \mathsf{F} \in \mathbb{B}$ simulieren läßt. Dabei sei $\mathbb{B}$ inklusive seiner kanonischen und nicht–kanonischen Elemente wie in Übung 5.4 definiert.

Die Korrektheit der Simulation von Void *durch* $\mathsf{T=F} \in \mathbb{B}$ *läßt sich am besten dadurch zeigen, daß man in den Inferenzregeln für* Void *letzteres durch die angegebene Simulation ersetzt und dann unter Zuhilfenahme der Regeln für* $\mathbb{B}$ *zeigt, daß jeweils die selben Unterziele entstehen wie in den ursprünglichen Regeln. Betrachten Sie dabei der Einfachheit halber nur die Regeln* `voidEq` *und* `voidE`*. Wie könnte man* `any(x)` *simulieren?*