

Improvements for Free

Daniel Seidel* und Janis Voigtländer, Universität Bonn
{ds,jv}@informatik.uni-bonn.de

Wir präsentieren eine Erweiterung der in “Theorems for free!” (Wadler, 1989) erstmals vorgestellten Theorie freier Theoreme, um relative Laufzeitaussagen. Freie Theoreme sind typbasierte Aussagen über Funktionen, d.h. allein vom Typ einer Funktion wird eine Aussage über das Verhalten der Funktion abgeleitet. Bisher waren solche Aussagen rein extensional: Werte von Programmfragmenten wurden in Beziehung gesetzt, während Eigenschaften wie Laufzeit oder Speicherbedarf außen vor gelassen wurden. Da freie Theoreme insbesondere zum Beweis der Korrektheit von Programmtransformationen herangezogen werden, sind Aussagen über Laufzeiten von großem Interesse.

Im Vortrag betrachten wir ein getyptes Lambda-Kalkül mit Typvariablen, welches als stark vereinfachter Teil einer Kernsprache funktionaler Programmiersprachen gesehen werden kann, und werden für Funktionen vom Typ $\alpha \rightarrow \text{Int}$ (polymorpher Typ nach Int) zunächst das bisher betrachtete freie Theorem ohne Laufzeitaussagen herleiten, um dann die notwendigen Anpassungen zum Erhalt eines freien Theorems mit Laufzeitaussage zu erläutern. Dabei gehen wir auf den Einfluss der Auswertungsstrategie (call-by-name vs. call-by-need vs. call-by-value) ein. Die formale Erweiterung freier Theoreme um Laufzeitaussagen findet bzgl. einer strikten (call-by-value) Sprache statt.

Für unser konkretes Beispiel, den Typ $\alpha \rightarrow \text{Int}$, liefert das (rein extensionale) freie Theorem, dass f eine konstante Funktion ist. Das konkrete Theorem lautet: Für alle Funktionen f des Typs $\alpha \rightarrow \text{Int}$ gilt, dass für beliebige Typen τ_1, τ_2 , und Funktionen g des Typs $\tau_1 \rightarrow \tau_2$, sowie beliebige Werte x vom Typ τ_1 ,

$$f\ x = f\ (g\ x) \tag{1}$$

Die entsprechende Erweiterung des Theorems liefert zusätzlich, dass die linke Seite von (1) effizienter ist als die rechte. Die Aussage ist nicht ganz trivial: Nur, da f für alle Eingaben identische Laufzeiten hat und in einer strikten Sprache, zusätzlich zu den Rechnungen auf der linken Seite, auf der rechten Seite die Anwendung von g auf x berechnet werden muss, gilt die Aussage.

Am Ende des Vortrags gehen wir kurz darauf ein, inwieweit nützliche relative Laufzeitaussagen bei komplizierteren Typen getroffen werden können, inwieweit Laufzeitkosten realistisch modelliert werden, und was bei der Untersuchung anderer Auswertungsstrategien zu beachten ist.

Die Ergebnisse wurden bereits auf dem “Workshop on Quantitative Aspects of Programming Languages (QAPL 2011)” vorgestellt und erscheinen in den Post-Proceedings.

*Der Autor wurde durch die DFG im Rahmen des Projektes VO 1512/1-1 gefördert.