

Erweiterung der Containerdarstellung von Listen mittels Relationen

Daniel Seidel¹ und Janis Voigtländer, Universität Bonn
{ds,jv}@iai.uni-bonn.de

Erweiterte Zusammenfassung

Für Beweise ist es im Allgemeinen hilfreich, von unnötigen Fakten zu abstrahieren. Betrachtet man Beweise über Eigenschaften polymorpher Funktionen auf Listen (Funktionen vom Typ $f :: [a] \rightarrow [a]$, z.B. *reverse*²), bedeutet dies, von den tatsächlichen Listenelementen zu abstrahieren.

Bundy and Richardson [1999] beschreiben eine abstrakte Listendarstellung, die heute als Containerdarstellung bekannt ist. Die Idee ist die Trennung von Struktur (Container) und Daten (Inhalt). Eine Liste kann als Paar (n, f) dargestellt werden, wobei n die Länge der Liste und f eine Abbildungen der Listenpositionen auf die dort gespeicherten Elemente bezeichnet.

Diese Definition eignet sich besonders um Beweise über polymorphe Funktionen zu führen. Z.B. lässt sich (für endliche Listen) sehr leicht zeigen, dass die zweifache Anwendung von *reverse* wieder die ursprüngliche Liste liefert. Wir definieren $reverse^c(n, f) = (n, \lambda i \rightarrow f(n - i - 1))$ als *reverse* auf Listencontainern und erhalten $reverse^c(reverse^c(n, f)) = reverse^c(n, \lambda i \rightarrow f(n - i - 1)) = (n, \lambda i \rightarrow f(n - (n - i - 1) - 1)) = (n, f)$.

Bereits bei Bundy and Richardson [1999] und auch bei Prince et al. [2008], wo die ursprüngliche Idee kategorientheoretisch betrachtet, präzisiert und auf andere Datentypen erweitert wird, werden Erweiterungen für Funktionen die nicht vollständig polymorph im Elementtyp sind, diskutiert. Betrachten wir eine Funktion *nub*, die doppelte Einträge einer Liste entfernt, so abstrahiert sie immer noch über den konkreten Elementtyp, verlangt aber einen Gleichheitstest auf den Listeneinträgen. Bei Sortierfunktionen wird eine Ordnungsrelation auf dem Elementtyp benötigt.

Die eben angegebenen Funktionen benötigen also mehr Informationen als die Struktur der Liste. Jedoch muss man nicht sofort die konkreten Listenelemente betrachten. Es genügt, die Containerdarstellung für Listen um die benötigte Relation zu erweitern — nicht als Relation zwischen konkreten Elementen, sondern zwischen Listenpositionen. Wir beschreiben diese Erweiterungen, zeigen Funktionen in ihrer Containerversion und stellen containerbasierte Beweise über Eigenschaften solcher Funktionen vor.

Literaturverzeichnis

- A. Bundy and J. Richardson. Proofs about lists using ellipsis. In *LPAR, Proc.*, volume 1705 of *LNCS*, pages 1–12. Springer-Verlag, 1999.
- R. Prince, N. Ghani, and C. McBride. Proving properties about lists using containers. In *FLOPS, Proc.*, volume 4989 of *LNCS*, pages 97–112. Springer-Verlag, 2008.

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

²Die Funktion *reverse* dreht eine Liste um.