

Ein striktes coalgebraisches Berechnungsmodell

Baltasar Trancón y Widemann
Technische Universität Berlin
bt@cs.tu-berlin.de

Zusammenfassung Die Semantik von freien Datentypen und reinen Funktionen wird meist durch eines von zwei gegensätzlichen Paradigmen definiert: *Fundierte* oder *nichtfundierte Algebra*. Die fundierte oder *strikte* Sichtweise zeichnet sich durch die vollständige Auswertung von Funktionsargumenten (*call-by-value*) und damit durch gut abschätzbare Aufwands- und Terminationsverhalten, sowie einfache Implementierung und weitgehend kontextunabhängige Codierung von Daten aus. Diese Vorteile werden erkauft durch den Verzicht auf *unendliche Daten*, wie sie in der imperativen und objektorientierten Welt in Form von *Iteratoren* und *zyklischen Graphen* allgegenwärtig sind.

Charakteristisch für den konträren, nichtfundierten Ansatz ist die Übergabe von unausgewerteten Berechnungen (*call-by-name*). Dadurch wird die Analyse von Aufwand und Termination ebenso wie die Kommunikation und das *marshalling* von Daten erheblich kompliziert. Als Ausgleich gewinnt man die Möglichkeit, unerhebliche Berechnungen zu unterdrücken (*laziness*) und somit endliche Anfangsstücke von konzeptuell unendlichen Daten und Abläufen zu verarbeiten. Zyklische Graphen werden in dieser Sichtweise als periodische unendliche Auffaltungen gehandhabt. Die Information, dass sich eine Berechnung geschachtelt wiederholt, ist jedoch *nicht zugänglich*.

Eine hybride Sichtweise, in der zyklische Daten nicht nur zulässig sind, sondern die Form der Zyklen auch nutzbar gemacht wird, ist die *Coalgebra*, das kategorielle Duale der Algebra. Diese Theorie ist stärker als die Algebra und hebt die Unterscheidung von Fundiertheit und Nichtfundiertheit auf. Diese zusätzliche Stärke wird durch eine explizit gegebene Identität von Ausdrücken erreicht. Anstelle der strukturellen Gleichheit wird die *Ununterscheidbarkeit des Verhaltens* gesetzt. Damit hat sich Coalgebra in den Gebieten Objektorientierung, Datenbanken, Automaten und Prozesse in den letzten Jahren als enorm elegante und leistungsfähige Theorie erwiesen. Mit Hilfe einer coalgebraischen Semantik lassen sich auf zyklischen Daten auch vollständig konsumierende Funktionen wie beispielsweise textuelle Darstellung (*pretty-printing*) oder Vergleiche auf endliche Berechnungen zurückführen, was algebraisch, auch ohne Fundiertheit, wegen der Nichtbeachtung von Identitäten nicht möglich ist, beziehungsweise Umcodierung in eine zyklensfreie Darstellung voraussetzt.

Der endlich darstellbare Anteil einer Coalgebra kann im Wesentlichen mit den gleichen rekursiven Techniken wie eine fundierte Algebra verarbeitet werden. Läuft die Berechnung in einen Zyklus, kann dieser anhand der Informationen auf dem Stack erkannt werden. Ist die zu berechnende Funktion so implementiert, dass das Ergebnis *top-down* erzeugt wird, also die Wurzel eines Teilergebnisses vor dem rekursiven Abstieg festgelegt und auf dem Stack verzeichnet werden kann, dann lässt sich der Zyklus durchbrechen, indem das Ergebnis des Anfangspunktes als Ergebnis des Endpunktes wiederverwendet wird und so wiederum einen Zyklus erzeugt. Funktionen mit dieser Eigenschaft entsprechen genau der Klasse der *primitiv corekursiven Funktionen*. Dabei kann die Erkennung und Behandlung von Zyklen in der Laufzeitumgebung vollständig und unabhängig von den darüber implementierten Funktionen realisiert werden. Mit denselben Mitteln lassen sich auch kleinste, größte und gemischte Fixpunkte zyklisch definierter Prädikate entscheiden.

Die naive Analyse des gesamten Stacks bei jedem Funktionsaufruf verursacht zusätzlichen, für die meisten Berechnungen inakzeptablen Laufzeitaufwand. Durch extensive statische und dynamische Optimierungen kann die teure Stackanalyse auf das für Termination notwendige Maß reduziert werden. Insbesondere auf de facto zyklensfreien Daten wird annähernd die Effizienz einer klassischen algebraischen Implementierung erreicht.

Das Prinzip der coalgebraischen Berechnungstechnik soll im Vortrag an einer experimentellen virtuellen Maschine demonstriert werden. Diese Maschine ist in ihrer imperativen Grundausstattung mit der Java-Maschine vergleichbar, verfügt aber außerdem über einen eingebauten Mechanismus zur Zyklusbehandlung. Als Beispiele sollen einfache, aber nichttriviale Berechnungen und Transformationen auf unendlichen periodischen Strömen dienen.