

The Algebra of Curry

Jan Christiansen Bernd Braßel
jac@informatik.uni-kiel.de bbr@informatik.uni-kiel.de

Christian-Albrechts-Universität zu Kiel

Zusammenfassung

In der Vergangenheit wurde gezeigt, dass die Relationen Algebra sich sehr gut für den Einsatz im Bereich der Semantik von Programmiersprachen eignet. Zunächst stellen wir die dafür notwendigen Grundlagen vor. Danach präsentieren wir unsere Erweiterung der vorhandenen Ideen, um moderne Konzepte funktional logischer Sprachen wie Pattern Matching, Laziness und Nicht-Determinismus mit Call-Time-Choice zu modellieren.

Dabei stellt sich heraus, dass sich eine relationale Semantik sehr gut für eine nicht-deterministische Sprache eignet, da Relationen eine intuitive Modellierung des Nicht-Determinismus erlauben. Die konkrete Relationen Algebra bietet außerdem eine sehr naheliegende Anschauung der Semantik als Teilmengen der kartesischen Produkte von Vor- und Nachbereich einer Funktion.

Das Ziel der vorgestellten Arbeit ist die Definition einer relationalen denotationellen Semantik für die Programmiersprache Curry. Aufbauend auf den Vortrag „Towards A New Denotational Semantics For Curry - A Pointless Presentation“ zeigen wir wie Curry Programme, sehr einfach in Relationen Algebra übersetzt werden können. Mit Hilfe dieser Semantik können wir zum Beispiel „Function-Patterns“ mittels Transposition eine sehr einfache Semantik geben.

Wir vermuten, dass diese Art der Semantik vielseitiger anwendbar ist als die vorhandenen denotationellen Semantiken für Curry. In zukünftigen Arbeiten wollen wir die Semantik verwenden, um eine Systematik für die Inversion von Funktionen zu entwickeln und zu verifizieren. Im Gegensatz zu Ansätzen in der rein funktionalen Programmierung kann dieses Konzept aufgrund des Nicht-Determinismus' in Curry direkt umgesetzt werden. Außerdem sehen wir Anwendungen im Bereich der Verifikation von Programmanalysen, die auf abstrakter Interpretation beruhen. Schließlich wollen wir auch die Äquivalenz zu vorliegenden operationalen Semantiken beweisen, um eine seit langem bestehende Lücke im Bereich der funktional logischen Programmiersprachen zu schließen.