Tracing Curry by Program Transformation

Frank Huch, Bernd Brassel und Michael Hanus
Institut für Informatik, CAU zu Kiel

Zum Debuggen deklarativer, insbesondere nicht-strikter funktionaler und funktional-logischer Sprachen hat sich gezeigt, daß der Standardansatz mit schrittweiser Programmausführung und Visualisierung der aktuellen Konfiguration nicht zur Fehlerfindung geeignet ist. Insbesondere durch die verzögerte Auswertung ist die Reduktionssemantik nur schwer nachzuvollziehen. Als ein möglicher alternativer Ansatz wurde *Tracing* vorgeschlagen, bei dem ein Programmlauf aufgezeichnet wird und dieser nach dem Programmende (hierbei sind auch Laufzeitfehler und Programmunterbrechung möglich) mit speziellen Tools analysiert werden kann. So kann die verzögerte Auswertung beispielsweise wie eine strikte Auswertung durchlaufen werden, wobei komplette Unterberechnungen wie bei Standarddebuggern für imperative Sprachen "geskipt" werden können. Auch sind andere Darstellungen, wie z.B algorithmisches Debuggen oder eine Bottom-Up-Sicht der Berechnungsergebnisse möglich.

Dieser Vortrag beschäftigt sich mit der Erzeugung solcher Traces für die nicht-strikte funktional-logische Sprache Curry. Neben einem interpreterbasierten Prototypen haben wir eine Programmtransformation entwickelt, mit der Curry Programme transformiert werden, so daß sie als Seiteneffekt einen Trace in eine Datei schreiben. Ähnliche Ansätze gibt es bereits im Hat-System für die Programmiersprache Haskell. Bei der Erweiterung auf Curry müssen insbesondere die logischen Features, wie nichtdeterministische Auswertung und logische Variablen, berücksichtigt werden.