

# Funktionaler Fauler Nichtdeterminismus

Sebastian Fischer

Christian-Albrechts Universität zu Kiel  
sebf@informatik.uni-kiel.de

Logisch-funktionale Programmierung zeigt, dass die Kombination von bedarfsgesteuerter mit nichtdeterministischer Auswertung weitreichende Vorteile zum Lösen von Suchproblemen hat. Diese Vorteile erschließen sich rein funktionalen Programmierern nur bedingt, da die existierende Unterstützung für bedarfsgesteuerte Auswertung und Nichtdeterminismus in funktionalen Sprachen nicht einfach kombiniert werden kann.

Ich stelle eine gemeinsame Arbeit mit Oleg Kiselyov und Chung-chieh Shan vor, in der wir eine—in die funktionale Programmiersprache Haskell eingebette—Sprache für nichtdeterministische Berechnungen mit bedarfsgesteuerter Auswertung entwickeln. Neben einer gleichungsbasierten Spezifikation des für logisch-funktionale Sprachen entwickelten Begriffes *Call-Time Choice*, der das Verhalten der genannten Kombination klärt, stelle ich Techniken vor, die es erlauben, diese Spezifikation effizient zu implementieren. Unsere Implementierung hat ein vergleichsweise hohes Abstraktionsniveau und erlaubt zum Beispiel die Integration beliebiger Suchstrategien. Sie bietet—neben ihrer Verwendbarkeit als rein funktionale Bibliothek—eine flexible Basis für die Übersetzung logisch-funktionaler Programme nach Haskell. Erste Vergleiche mit existierenden Übersetzern für die logisch-funktionale Programmiersprache Curry sind vielversprechend.