

# Muli: Constraint-Programmierung in Java auf symbolischer JVM

Jan C. Dageförde und Herbert Kuchen, WWU Münster

*Muli Lang* (Münster Logic-Imperative Language) ergänzt die Programmiersprache Java um Möglichkeiten zur vereinfachten Lösung von Suchproblemen. Durch Java als Grundlage bietet Muli Lang gewohnte und geschätzte Features objektorientierter Programmiersprachen, wie Vererbung und Kapselung von Verhalten und Struktur. Da in der (u. a. betriebswirtschaftlichen) Praxis Suchprobleme auftreten, für deren Lösung Java-Programmierung in der Regel unhandlich ist, führt Muli Lang durch minimale sprachliche Ergänzungen logische Variablen, symbolische Ausführung und eingekapselte Suche ein. Durch die enge Integration von Constraint-Programmierung und objektorientierter Programmierung ist Muli Lang insbesondere für diejenigen Suchprobleme interessant, die im Zeitverlauf inkrementell um Constraints ergänzt werden.

Muli Lang wird mit der Laufzeitumgebung *Muli Env* (Münster Logic-Imperative Environment) kombiniert, einer symbolischen Java Virtual Machine (SJVM), welche Muli- und Java-Programme sowohl symbolisch als auch regulär zur Ausführung bringt. Muli Env erweitert die übliche Java Virtual Machine (JVM) um Komponenten wie Choice Points und Trail, wie sie von abstrakten Maschinen für logische Programmiersprachen, wie der Warren Abstract Machine (WAM), bekannt sind. Dadurch wird Nichtdeterminismus in Muli-Programmen, in Form eingekapselter Suche und Backtracking, ermöglicht. Außerdem verwaltet Muli Env derzeit zwei Constraint Solver, welche die Suche nach spezifischen Lösungen übernehmen.

Die Auslagerung von nichtdeterministischer Ausführung und Suche an externe Programme, beispielsweise an spezialisierte Prolog-Programme, sowie die Einbettung von Constraint Solving-Laufzeitbibliotheken, entfallen so vollständig. Stattdessen werden diese Aufgaben nahtlos von der virtuellen Maschine übernommen.