

Aufgabenstellung

Ausgehend von einem Sprachmodul für funktionale Sprachen ist ein experimenteller Übersetzer zu entwerfen und zu implementieren, der die Sprache in einen Zwischencode übersetzt, der durch die G-Maschine interpretierbar ist.

Die Übersetzungskonzepte sind zu begründen und ihre Funktionsfähigkeit nachzuweisen.

ELCOM — Ein Compiler für eine funktionale Programmiersprache.
Großmann, Dirk. — 1991. — 191 S., 20 Abb., 17 Lit.
Technische Universität Chemnitz, Fachbereich Informatik. Diplomarbeit.

Kurzreferat

Diese Arbeit beschäftigt sich mit dem Entwurf und der Implementation eines Compilers für eine Version des getypten erweiterten Lambda-Kalküls mit verzögerter Auswertung. Dem Compiler liegt das Prinzip der trägen Superkombinator-Graphenreduktion zugrunde. Er übersetzt die Quellsprache in G-Code, der von der G-Maschine, einer abstrakten Maschine zur Graphenreduktion, abgearbeitet werden kann.

Das Ergebnis ist der funktionsfähige Compiler „ELCOM“, der unter dem Betriebssystem UNIX läuft und in *C* geschrieben ist. Der Quelltext ist als Anhang beigelegt.

Da man die einzelnen Übersetzungsphasen des Quellprogramms gut verfolgen kann, ist der Compiler zur Demonstration der verwendeten Methoden geeignet.

Thesen

1. Die funktionalen Programmiersprachen haben eine Reihe von Vorteilen gegenüber den prozeduralen Sprachen. Funktionale Sprachen haben eine einfache und klare Semantik. Ein funktionales Programm ist wesentlich kürzer als ein gleichartiges, das in einer prozeduralen Sprache geschrieben ist. Die den funktionalen Sprachen innewohnende Parallelität macht sie geeignet zur Ausnutzung der Möglichkeiten von Mehrprozessorrechnern.
2. Funktionale Sprachen mit verzögerter Auswertung sind ein semantischer Fortschritt gegenüber den strikten Sprachen.
3. Alle funktionalen Sprachen lassen sich in den erweiterten Lambda-Kalkül transformieren.
4. Einer breiten Anwendung der funktionalen Programmierung steht die Tatsache im Wege, daß funktionale Programme langsamer ausgeführt werden als ihre prozeduralen Gegenstücke. Schnellere Implementationen können die Akzeptanz der funktionalen Sprachen bei den Anwendern erhöhen.
5. Die träge Graphenreduktion ist eine adäquate Abarbeitungsstrategie für Programme, die in funktionalen Sprachen mit verzögerter Auswertung geschrieben sind.
6. Die Überführung eines funktionalen Programms in Superkombinator-Form macht es möglich, aus jeder Funktion eine feste Code-Folge zu erzeugen. Durch die Superkombinatoren wird das Programm nicht in zu winzige Abarbeitungsschritte zerlegt.
7. Die G-Maschine ist eine extrem schnelle Implementation der verzögerten Superkombinator-Graphenreduktion.
8. Der in dieser Arbeit entwickelte Compiler „ELCOM“ erzeugt optimierten G-Code. Er ist gut zur Demonstration der verwendeten Methoden geeignet.

Erklärung

Hiermit erkläre ich, daß ich diese Arbeit selbständig und nur unter Verwendung der aufgeführten Literatur angefertigt habe.

Chemnitz, den 1. Juli 1991

Dirk Großmann