Informatik-Fachberichte 210

Herausgeber: W. Brauer
im Auftrag der Gesellschaft für Informatik (GI)

Klaus Drosten

Termersetzungssysteme

Grundlagen der Prototyp-Generierung algebraischer Spezifikationen

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo

Autor

Klaus Drosten
Gesellschaft für Mathematik und Datenverarbeitung (F4)
Dolivostraße 15, D-6100 Darmstadt

CR Subject Classification (1987): F.4.1, I.2.2-3, D.2.m

ISBN-13:978-3-540-51172-4 e-ISBN-13:978-3-642-74769-4
DOI: 10.1007/978-3-642-74769-4

CIP-Titelaufnahme der Deutschen Bibliothek.
Drosten, Klaus:
Termersetzungssysteme: Grundlagen der Prototyp-Generierung algebraischer Spezifikationen / Klaus Drosten. – Berlin; Heidelberg; New York; London; Paris; Tokyo: Springer, 1989
(Informatik-Fachberichte; 210)
ISBN-13:978-3-540-51172-4 (Berlin ...) brosch.

NE: GT

2145/3140 – 543210 – Gedruckt auf säurefreiem Papier

Vorwort

Die logische Programmierung geht auf den Anfang der 70er Jahre zurück und wurde besonders populär in Verbindung mit der Programmiersprache PROLOG. Ein Hauptmerkmal der logischen Programmierung ist die strikte Trennung von Kontrollfluß und problemspezifischem Wissen. Alle problemspezifischen Aussagen werden in einer "Wissensbank" zusammengefaßt und mit Hilfe einer problemunabhängigen Inferenzmaschine ausgewertet. Die algebraische Programmierung entstand in ihren Grundzügen Mitte der 70er Jahre, als die ersten Arbeiten über die Spezifikation abstrakter Datentypen veröffentlicht wurden. Obwohl sie auf gemeinsamen Prinzipien beruhen, entwickelten sich die Gebiete der algebraischen und der logischen Programmierung zunächst unabhängig voneinander. Erst in jüngster Zeit wurde mit Erfolg versucht, beide Ansätze in einem gemeinsamen Kalkül zu vereinen.

Die vorliegende Arbeit ist in fünf Kapitel gegliedert. Kapitel 1 gibt einen ausführlichen Überblick über die Grundlagen der Ausführung algebraischer Spezifikationen. Kapitel 2 enthält eine Zusammenfassung der benötigten Grundbegriffe und dient als Vorbereitung auf den anschließenden Hauptteil der Arbeit. In Kapitel 3 und 5 wird das Grundkonzept der algebraischen Spezifikation um Ausdrucksmittel zur Fehlerbehandlung und Modularisierung in abstrakten Datentypen erweitert. Die Ausdrucksmittel werden besonders im Hinblick auf ihre Operationalisierbarkeit untersucht. In Kapitel 4 wird schließlich gezeigt, wie (und wann) sich algebraische Spezifikationen automatisch in PROLOG-Programme übersetzen und mit deren Hilfe ausführen lassen.

Das Buch wendet sich an alle, die an den theoretischen Grundlagen der algebraischen und logischen Programmierung interessiert sind. Zum besseren Verständnis des Textes sind Grundkenntnisse in der mathematischen Logik hilfreich und wünschenswert. Als Einstieg wird eines der zahlreichen Lehrbücher über die Grundlagen des automatischen Beweisens empfohlen. Weiterführende Literatur kann der ausführlichen Bibliographie am Ende dieser Arbeit entnommen werden.

Das Buch entstand als Überarbeitung einer Dissertation, die ich 1987 an der TU Braunschweig angefertigt habe. Wesentliche Erkenntnisse, die für die Erstellung der Dissertation relevant waren, habe ich während meiner Arbeit an einem von der Deutschen Forschungsgemeinschaft geförderten Projekt über die Spezifikation abstrakter Datentypen erworben. Geleitet wurde das Projekt von Prof. Dr. H.-D. Ehrich, für dessen Unterstützung ich mich an dieser Stelle ausdrücklich bedanken möchte. Mein weiterer Dank gilt Frau H. Huschka, die den wesentlichen Teil des Textes mit Hilfe des Textverarbeitungssystems SIGNUM auf einem Atari PC eingegeben hat.

Darmstadt, im Februar 1989 Klaus Drosten

Zusammenfassung

Termersetzungssysteme sind ein nicht-deterministisches Berechnungsmodell aus dem Bereich der funktionalen Programmierung. Die Funktionen werden durch rekursive Regeln spezifiziert und durch Untertermersetzung ohne explizite Kontrolle ausgewertet. Die Semantik eines Termersetzungssystems ist wohldefiniert, wenn alle Terme zu Normalformen reduzierbar sind (Termination) und das Ergebnis der Berechnung nicht von der Wahl der Regeln abhängt, die zur Reduktion benutzt werden (Konfluenz).

Das Prinzip der Termreduktion basiert auf der Idee, Terme durch gezielte Anwendung von Gleichungen in einfachere Ausdrücke umzuformen. Die dabei erzeugten Reduktionsfolgen entsprechen Herleitungen im Gleichungskalkül, bei denen die Reflexivitäts- und Symmetrieregeln überhaupt nicht und die Transitivitätsregel nur am Schluß angewandt werden. Für Gleichungsmengen, die terminierend und konfluent sind, ist die Termreduktion eine konsistente und vollständige Inferenzregel, da alle logischen Folgerungen allein durch die Berechnung von Normalformen herleitbar sind. Gleiches gilt in einer mehrsortigen Logik unter der Annahme, daß die Individuenbereiche in allen Modellen nicht-leer sind.

Die Methode der algebraischen Spezifikation geht von der Beobachtung aus, daß Datentypen Algebren sind, und bietet als Hauptinstrument Gleichungen an, um Algebren unabhängig von einer konkreten Repräsentation der Daten zu beschreiben ("abstrakter Datentyp"). Der enge Zusammenhang von Ersetzungsregeln und Gleichungen bildet dabei das theoretische Fundament, um Termersetzungssysteme als Werkzeug zur Prototyp-Generierung algebraischer Spezifikationen einzusetzen. Allerdings sind konventionelle Termersetzungssysteme nicht mächtig genug, die Aspekte der Fehlerbehandlung und Modularisierung in abstrakten Datentypen zu berücksichtigen; sie werden deshalb in der vorliegenden Arbeit um Ausdrucksmittel zur Parametrisierung und zur Einschränkung von Variablenbereichen erweitert. Die eigentliche Ausführung algebraischer Spezifikationen wird durch die Übersetzung von Termersetzungssystemen in PROLOG-Programme unterstützt. Im Überblick stellen sich die einzelnen Punkte wie folgt dar:

Termersetzungssysteme mit eingeschränkten Variablen subsumieren das Konzept der ok- und unsicheren Variablen, das ein geeignetes Beschreibungsmittel für Fehler- und Ausnahmesituationen in abstrakten Datentypen ist. Die konventionellen Methoden zur Überprüfung der Konfluenz und Termination werden auf eingeschränkte Variablen verallgemeinert und bilden die Basis zur Ausführung von Fehlerspezifikationen. Obwohl Einschränkungen von Variablenbereichen auch mit Hilfe von partiellen Sorten und überladenen Operatoren modelliert werden können, bietet das Konzept der eingeschränkten Variablen im Hinblick auf die Operationalisierung algebraischer Spezifikationen einige Vorteile.

Auf der Grundlage parametrischer Termersetzungssysteme wird untersucht, ob und inwieweit sich einzelne, ausführbare Module automatisch zu einer komplexen, ausführbaren Gesamtspezifikation zusammensetzen lassen. Das Kompositionstheorem gibt eine positive Antwort für den Fall, daß die formalen Parameter der beteiligten Module nur aus Sorten bestehen und die Parameterübergabe-Morphismen injektiv sind. Wann immer das Kompositionstheorem anwendbar ist, kann der globale Test auf Ausführbarkeit eines Gesamtsystems durch eine Reihe lokaler Tests auf Modulebene ersetzt werden.

Die Übersetzung von Termersetzungssystemen in PROLOG-Programme ist ein eleganter Ansatz, um die automatische Ausführung algebraischer Spezifikationen zu unterstützen. Auf der Basis einer "brute force" Strategie ist es prinzipiell möglich, jedes wohldefinierte Termersetzungssystem in ein äquivalentes PROLOG-Programm zu übersetzen. Für praktische Anwendungen ist ein solches Vorgehen jedoch ungeeignet, da ein Durchsuchen des Reduktionsbaumes der Breite nach nicht nur einen immensen Bedarf an Speicherplatz verursacht, sondern auch zu einem inakzeptablen Laufzeitverhalten führt. Effiziente Programme können nur erzeugt werden, wenn der Reduktionsbaum der Tiefe nach durchsucht wird. Als Terminationskriterium für Tiefensuchmethoden dient die Annahme, daß keine unendlichen Reduktionsfolgen existieren.

Inhaltsverzeichnis