
Vorwort

Übersetzer für höhere Programmiersprachen sind große komplexe Softwaresysteme. Sie haben aber einige besondere Eigenschaften, die sie vor den meisten anderen Softwaresystemen auszeichnen.

Ihre Funktionalität ist (fast) wohldefiniert. Idealerweise existieren vollständige formale oder zumindest präzise Beschreibungen der Quellsprache und der Zielsprache. Häufig gibt es dazu noch Beschreibungen von Schnittstellen zum Betriebssystem, zum Programmiersystem und zu Programmierungsumgebungen, zu anderen Übersetzern und zu Programmbibliotheken.

Die Übersetzungsaufgabe lässt sich auf natürliche Weise in Teilaufgaben zerlegen. Diese Zerlegung ergibt eine modulare Struktur, welche übrigens auch die Struktur der meisten Übersetzerbaubücher induziert.

In den fünfziger Jahren wurde bereits erkannt, dass die Implementierung von Anwendungssystemen direkt in der Maschinensprache sowohl mühsam als auch fehleranfällig ist und zu Programmen führt, die genauso schnell veralten wie die Rechner, für die sie entwickelt wurden. Mit der Erfindung höherer, maschinenunabhängiger Programmiersprachen ergab sich aber auch sofort die Notwendigkeit, Übersetzer bereitzustellen, die in der Lage sind, Programme dieser höheren Programmiersprachen in Maschinensprache zu übersetzen.

Deshalb sind seit den fünfziger Jahren die verschiedenen Teilaufgaben der Übersetzung Gegenstand intensiver Forschung. Für die Teilaufgabe der syntaktischen Analyse von Programmen wurden etwa Konzepte aus der Theorie der formalen Sprachen und der Automaten wie reguläre Ausdrücke und endliche Automaten, kontextfreie Grammatiken und Kellerautomaten übernommen und im Hinblick auf die gegebene Anwendung weiterentwickelt. Die Bearbeitung der Problemstellung war dabei so erfolgreich, dass die Realisierung der für syntaktische Analyse benötigten Komponenten (fast) vollständig automatisiert werden konnte: anstatt „zu Fuß“ implementiert zu werden, werden diese Komponenten heute häufig aus Spezifikationen, in diesem Fall kontextfreien Grammatiken, generiert. Solche Generierungsverfahren werden auch für andere Komponenten eines Übersetzers angestrebt und sind zum Teil bereits realisiert.

Das vorliegende Buch strebt nicht an, ein Kochbuch für Übersetzer zu sein. Man wird hier darum keine Rezepte finden der Art: „Um einen Übersetzer von

der Quellsprache X in die Maschinensprache Y zu konstruieren, nehme man. . .“ Unsere Darstellung reflektiert dagegen die oben aufgezählten Besonderheiten des Übersetzerbaus, insbesondere das Vorhandensein präziser Aufgabenstellungen und das Bestreben, diese genau zu verstehen und angemessene theoretische Konzepte zu ihrer systematischen Behandlung bereitzustellen. Idealerweise können solche Konzepte die Grundlage automatischer Generierungsverfahren bilden.

Dieses Buch ist für Studierende der Informatik bestimmt. Die Kenntnis zumindest einer imperativen Programmiersprache wird vorausgesetzt. Vorkenntnisse über formale Sprachen und Automaten sind nützlich, aber nicht unbedingt erforderlich. Das Buch enthält die notwendige Theorie.

Aufbau des Buches

Für die neue Auflage des Buchs, *Wilhelm/Maurer: Übersetzerbau*, entschieden wir uns, den Inhalt auf mehrere Bände zu verteilen. In einem ersten Band, *Wilhelm/Seidl: Übersetzerbau — Virtuelle Maschinen* wird beschrieben, was ein Übersetzer tut, also welche Korrespondenz zwischen einem Quellprogramm und einem Zielprogramm er herstellt. Dazu wird für eine imperative, eine funktionale, eine logische und eine objektorientierte Programmiersprache je eine geeignete *virtuelle* Maschine (in den früheren Auflagen *abstrakte* Maschine genannt) angegeben und die Übersetzung von Programmen der jeweiligen Quellsprache in die Sprache der zugehörigen virtuellen Maschine genau beschrieben.

Die weiteren drei Bände beschreiben, wie der Übersetzungsprozess strukturiert und organisiert ist. Sie korrespondieren jeweils zu einer Phase des Prozesses, *syntaktische und semantische Analyse*, *Programmtransformation* und *Codeerzeugung*. Im dem vorliegenden Band wird die Analysephase der Übersetzung beschrieben, realisiert durch das so genannte Frontend des Übersetzers. Wir werden die Frage behandeln, wie man den Übersetzungsprozess in einzelne Phasen unterteilt, welche Aufgaben dabei die einzelnen Phasen erledigen sollen, welche Techniken man in ihnen benutzt, wie man formal beschreiben kann, was sie tun, und wie eventuell aus einer solchen Beschreibung automatisch eine Übersetzerkomponente erzeugt werden kann.

Danksagung

Neben den Mitstreitern der früheren Auflagen möchten wir Michael Jacobs für seine Mitwirkung beim Kapitel Syntaxanalyse und ihm und Jörg Herter für sorgfältiges Korrekturlesen danken. Bei der Neuformulierung des rekursiven Abstiegsparsers hat uns schließlich Christoph Mallon entscheidend unterstützt.

Einstweilen wünschen wir aber der geneigten Leserin und dem geneigten Leser viel Vergnügen mit dem vorliegenden Band und hoffen, dass das Buch Appetit machen möge, für die Lieblings-Programmiersprache blitzschnell einen eigenen Übersetzer zu basteln.

Saarbrücken und München, April 2012

Reinhard Wilhelm, Helmut Seidl
und Sebastian Hack

Weitere Materialien zu diesem Buch finden Sie auf der Internet-Seite:
<http://www2.informatik.tu-muenchen.de/seidl/compilers/>

Übersetzerbau

Band 2: Syntaktische und semantische Analyse

Wilhelm, R.; Seidl, H.; Hack, S.

2012, XII, 247 S., Softcover

ISBN: 978-3-642-01134-4