

Übersetzer

Vorlesung WS 1997/98 Uwe Kastens

Ziele:

- grundlegende Verfahren der Sprachimplementierung erlernen,
- den Einsatz von **generierenden Werkzeugen** und von **Standardverfahren** kennen und schätzen lernen,
- Übersetzerkonstruktion als Beispiel für das Zusammenwirken von grundlegenden **Algorithmen**, **Theorien** und von **Software-Engineering-Methoden** bei **präzise definierten Aufgaben** erkennen,
- Anwendbarkeit der Verfahren auch für **Sprachen jenseits der Programmiersprachen** erkennen.

Inhalt

1. Übersetzeraufgaben und -struktur

Aufgabenzerlegung mit Bezug zu Sprachdefinitionen, Symbole, Syntax, statische und dynamische Semantik

2. Übersetzerstruktur, Schnittstellen

zentrale Datenstrukturen
Interpreter, Debugger

3. Lexikalische Analyse

Spezifikation und Implementierung endlicher Automaten, Generatoren dafür
Module zur Speicherung und Codierung von Symbolen

4. Syntaktische Analyse

Ziel- und quellorientierte Verfahren,
Rekursiver Abstieg, LR-Verfahren, Generatoren dafür,
Grammatikklassen, Fehlerbehandlung

5. Semantische Analyse

Bezeichneridentifikation, Typprüfung,
attributierte Grammatiken, Generatoren dafür

6. Optimierende Programmtransformation

Datenflußanalyse spezifiziert durch Gleichungssysteme,
Lösungsalgorithmen, Transformationen

7. Code-Erzeugung

Typische Eigenschaften von Zielprozessoren,
Verfahren zur Registerzuteilung
Optimierende Code-Auswahl, Generatoren dafür

8. Code-Parallelisierung

parallele Funktionseinheiten, Pipelines, Abhängigkeitsgraphen

Literatur

U. Kastens:

Übersetzerbau, Handbuch der Informatik 3.3,
Oldenbourg, 1990
(Hörscheine für Rabatt beim Kauf)

W. M. Waite, L. R. Carter:

An Introduction to Compiler Construction,
Harper Collins, New York, 1993

W. M. Waite, G. Goos:

Compiler Construction,
Springer-Verlag, 1983

R. Wilhelm, D. Maurer:

Übersetzerbau - Theorie, Konstruktion, Generierung,
Springer-Verlag, 1992

A. Aho, R. Sethi, J. D. Ullman:

Compilers - Principles, Techniques and Tools,
Addison-Wesley, 1986

A. W. Appel:

Modern Compiler Implementation in C,
Cambridge University Press, 1997
auch in Java und in ML

Vorlesung Übersetzer WS 97/98 / Folie 2

nachlesen:

Kastens / Übersetzerbau, Abschnitt Anhang enthält weitere Literatur.

Bemerkungen:

Hörscheine für das Buch gibt es in F2.317

Was übersetzt ein Übersetzer?

Ein **Übersetzer** transformiert korrekte Sätze einer **Quellsprache** in **gleichbedeutende** Sätze einer **Zielsprache**.

Quellsprache:	Zielsprache:
Programmiersprache	Maschinensprache
C++	Sparc-Code
Programmiersprache	Abstrakte Maschine
Java	Java-Byte-Code
Programmiersprache	Programmiersprache
Source-to-Source	
C++	C
Anwendungssprache	Anwendungssprache
LaTeX	HTML
DB-Sprache (SQL)	DB-Systemaufrufe

Ein **Interpreter** führt Sätze einer Quellsprache aus.

Lisp
Shell-Kommandosprache
Java-Byte-Code

Vorlesung Übersetzer WS 97/98 / Folie 3

Ziele:

- Vielfalt des Einsatzes von Übersetzern erkennen
- Abgrenzung zu Interpretieren

in der Vorlesung:

- Beispiele für verschiedene Quell-/Zielsprachen-Paare

nachlesen:

Kastens / Übersetzerbau, Abschnitt 1.

Übungsaufgaben:

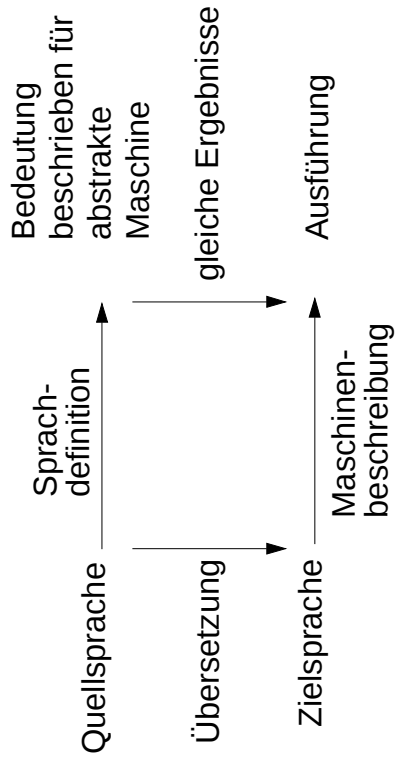
- Weitere Beispiele für Anwendungssprachen finden (Vorlesung)
- Aufgabe 3 Strukturen in Zielprogrammen zu einfachen Quellprogrammen erkennen

Verständnisfragen:

Nennen Sie Gründe für die Übersetzung in andere als Maschinensprachen.
Warum wird Lisp meist interpretiert statt übersetzt?

Bedeutungserhaltende Übersetzung

Ein **Übersetzer** transformiert **korrekte Sätze** einer **Quellsprache** in **gleichbedeutende Sätze** einer **Zielsprache**.



Die **Bedeutung** ist nur für **korrekte** Programme definiert.
Aufgabe: Fehlerbehandlung.

Der Übersetzer ermittelt **statische** Programmeigenschaften zur **Übersetzungszeit**, z. B. **übersetzbar**.

Dynamische Eigenschaften werden zur **Laufzeit** des Programms ermittelt, z. B. **ausführbar**.

Vorlesung Übersetzer WS 97/98 / Folie 4

Ziele:

- Definition von Quellsprache und Zielmaschine spezifizieren die Übersetzung.
- Begriffe Übersetzungszeit, Laufzeit

nachlesen:

Kastens / Übersetzerbau, Abschnitt 1.1

Sprachdefinition - Übersetzeraufgaben

• Notation der Grundsymbole

(Wortsymbole, Bezeichner, Literale)

formal definiert: reguläre Ausdrücke

Aufgabe: lexikalische Analyse

• syntaktische Struktur

formal definiert: kontextfreie Grammatik

Aufgabe: syntaktische Analyse

• statische Semantik

(Definition von Namen für Objekte, Typregeln)

meist verbal definiert

Aufgaben: semantische Analyse, Transformation

• dynamische Semantik

(Bedeutung, Wirkung von Sprachkonstrukten)

meist verbal definiert, in Termen einer abstrakten Sprachmaschine

Aufgaben: Transformation, Code-Generierung

• Definition der Zielsprache (-maschine)

Aufgaben: Transformation, Code-Generierung, Assemblierung

Vorlesung Übersetzer WS 97/98 / Folie 5

Ziele:

- Eigenschaften von Sprachkonstrukten einordnen
- typische Beschreibungsformen kennen

in der Vorlesung:

- Beispiele zu Sprachkonstrukten aus Sprachdefinitionen

nachlesen:

Kastens / Übersetzerbau, Abschnitt 1.2

Übungsaufgaben:

- Aufgabe 1 Fehlermeldungen provozieren
- Aufgabe 2 bestimmte Spracheigenschaften einordnen

Verständnisfragen:

Manche Sprachdetails können unterschiedlich eingeordnet werden. Diskutieren Sie für hypothetische Sprachen:

- "Array als Parameter verboten" - Syntax oder statische Semantik?
- "Indexbereich von Arrays nicht leer" - statische oder dynamische Semantik?

U-6 Übersetzeraufgaben		
Strukturierung	Lexikalische Analyse	Symbol-erkennung Konversion
	Syntaktische Analyse	Zerteilung Baumaufbau
Translation	Semantische Analyse	Bezeichner-analyse Typ-analyse
	Transformation	Daten-abbildung Operations-abbildung
Kodierung	Code-Generierung	Ausführungs-reihenfolge Register-zuteilung Instruktions-Auswahl
	Assemblierung	Instruktions-kodierung Interne Adressierung Externe Adressierung

Vorlesung Übersetzer WS 97/98 / Folie 6

Ziele:

- Konsequente Entwicklung der Übersetzerstruktur aus den Übersetzeraufgaben
- in der Vorlesung:
 - Erläuterung der Aufgaben in der rechten Spalte
 - Zuordnung zu Abschnitten der Vorlesung

nachlesen:

Kastens / Übersetzerbau, Abschnitt 2.1

Zielmaschine - Übersetzaufgaben

Eigenschaften von Zielprozessoren	Aufgaben der Kodierung
Instruktionssatz RISC - CISC	Instruktionsauswahl
Instruktionsformat 2-, 3-Adress	Instruktionskodierung Registerzuteilung
Register Klassen, Anzahl,	Registerzuteilung Ausführungsreihenfolge
Speicheradressierung Ausrichtung, Adressbildung	Datenabbildung interne Adressierung Instruktionsauswahl Instruktionskodierung
Instruktionsausführung Parallelverarbeitung	Ausführungsreihenfolge

Vorlesung Übersetzer WS 97/98 / Folie 7

Ziele:

- Eigenschaften von Prozessoren nach Einfluß auf Übersetzaufgaben klassifizieren

in der Vorlesung:

- Beispiele für Zusammenhänge zwischen Prozessoreigenschaften und Übersetzaufgaben

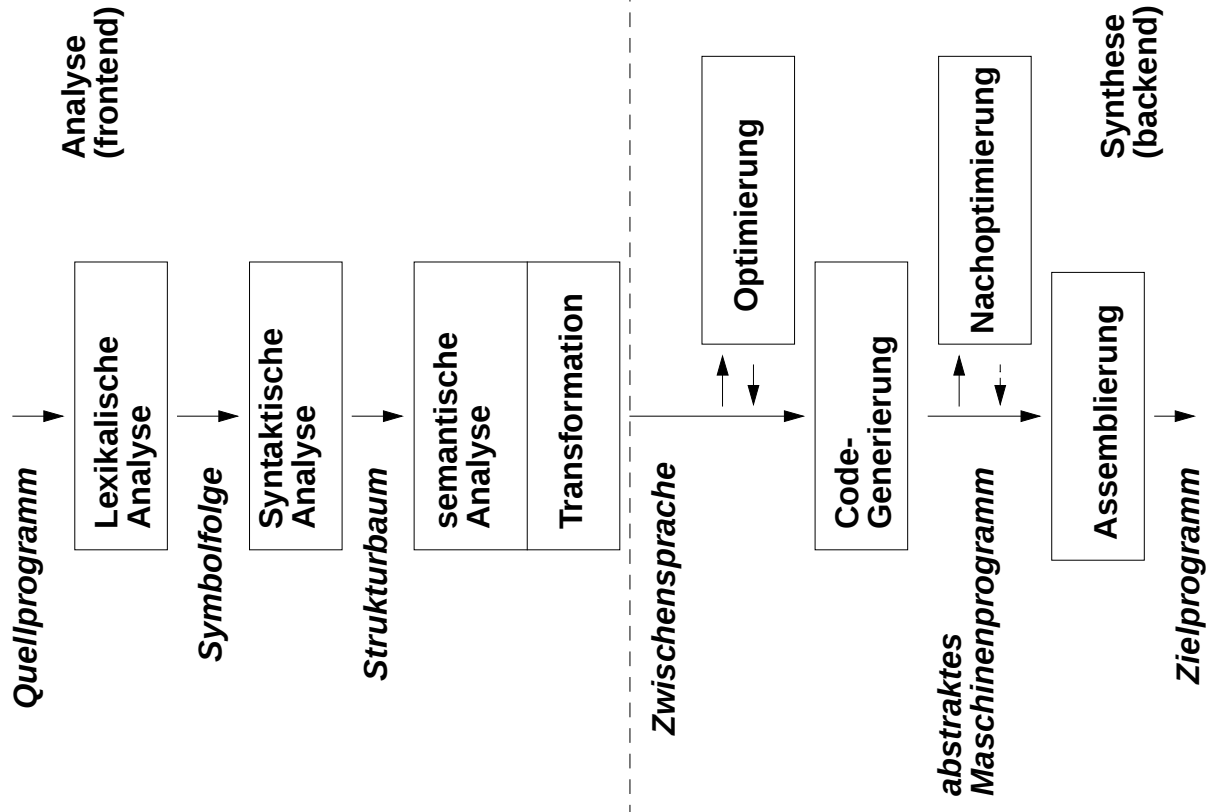
nachlesen:

Kastens / Übersetzerbau, Abschnitt 1.3

Verständnisfragen:

- Was hat die Ausführungsreihenfolge mit Registern zu tun?
- Könnte ein Übersetzer, der Instruktions-Pipelining des Prozessors ignoriert, trotzdem korrekt sein?
- Wovon werden die nicht genannten Aufgaben der Kodierung bestimmt?

Übersetzerstruktur und Schnittstellen



Vorlesung Übersetzer WS 97/98 / Folie 8

Ziele:

- Schnittstellen und Rollen der zentralen Phasen kennenlernen

in der Vorlesung:

- Beispiele zu den Schnittstellen
- Diskussion der Verständnisfragen

nachlesen:

Kastens / Übersetzerbau, Abschnitt 2.1

Ordnen Sie die verfeinerten Abb. 2.1-2, 2.1-3 den Übersetzeraufgaben auf U6 zu

Übungsaufgaben:

- Geben Sie mindestens fünf verschiedene Repräsentationen von Bäumen an.

Verständnisfragen:

- Es gibt Übersetzer, in denen manche der Schnittstellen konzeptionell - aber nicht als Datenstruktur existieren. Wie kommunizieren dann benachbarte Phasen?
- Welche Konsequenz hat das auf die zu lösenden Aufgaben?
- Geben Sie Beispiele für Rückkopplung.

Übersetzerqualitäten

- **Korrektheit:**
richtige Programme richtig übersetzen,
falsche Programme zurückweisen
- **Effizienz:**
Speicher- und Zeitbedarf des Übersetzers
- **Code-Effizienz:**
Speicher- und Zeitbedarf des Ziel-Codes
Aufgabe: Optimierung
- **Benutzerfreundlichkeit:**
Aufgabe: Fehlerbehandlung
- **Robustheit:**
jede Eingabe wird bewältigt

Strategien zur Übersetzerkonstruktion

- Sprachdefinition exakt befolgen
- generierende Werkzeuge einsetzen
- Standard-Module verwenden
- Standard-Methoden anwenden
- Validieren (Test-Suite)
- Verifizieren

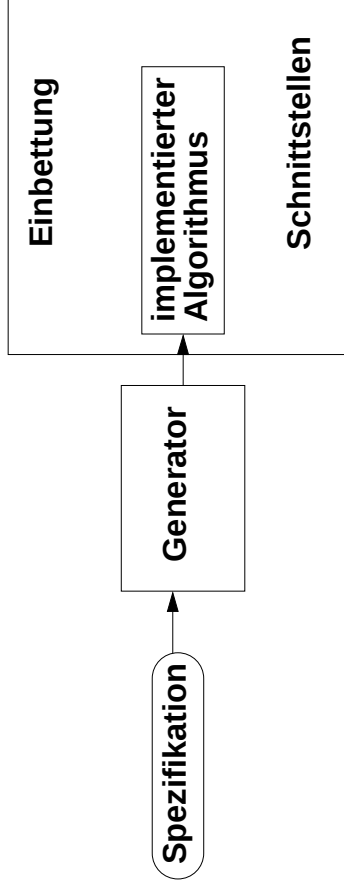
Ziele:

Übersetzer als Software-Produkt:

- Software-Qualitäten anstreben.
- Software-Engineering-Methoden anwenden.

Generatoren

Schema:



typische Generatöraufgaben:

reg. Ausdrücke	Scanner-Generator	endl. Automat
kontextfreie Gram.	Parser-Generator	Kellerautomat
attributierte Gram.	Attributauswert-Generator	Baumdurchlauf Berechnungen
Code-Muster	Code-Auswahl-Generator	Pattern-Matching

integriertes System Eli:



Vorlesung Übersetzer WS 97/98 / Folie 10

Ziele:

- Übersicht über Generator Einsatz

in der Vorlesung:

- Beispiele zu Generatoren

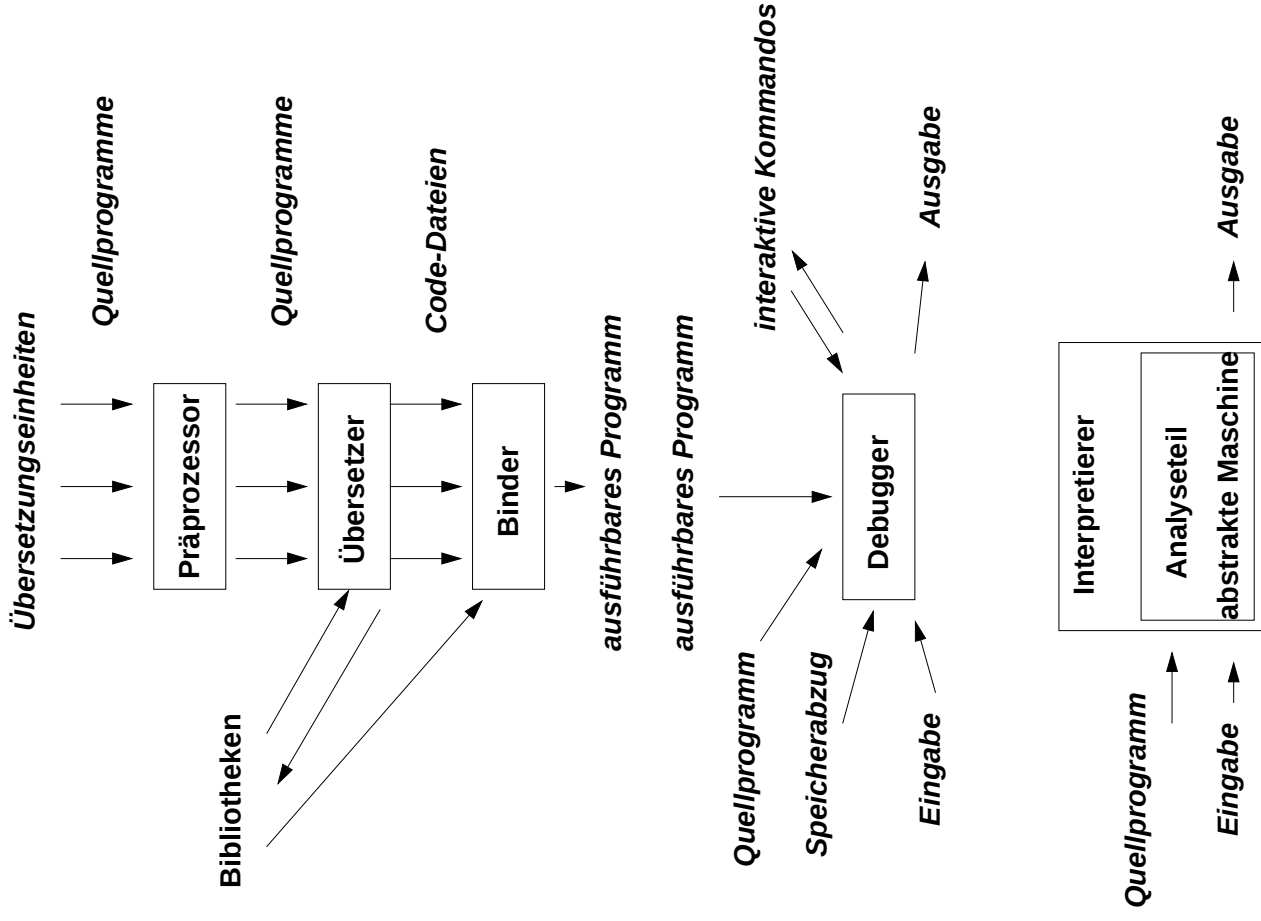
nachlesen:

Kastens / Übersetzerbau, Abschnitt 2.5

Übungsaufgaben:

- möglichst alle generierenden Werkzeuge des Eli-Systems finden

Umgebung des Übersetzers



Vorlesung Übersetzer WS 97/98 / Folie 11

Ziele:

- Zusammenwirken des Übersetzers mit anderen Sprachwerkzeugen verstehen.

in der Vorlesung:

- Erläuterung der Rollen der Sprachwerkzeuge und der Datenflüsse.

nachlesen:

Kastens / Übersetzerbau, Abschnitt 2.4