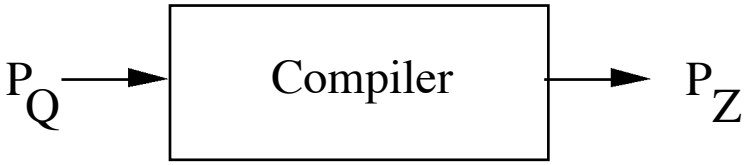
Kapitel 1: Grundlagen

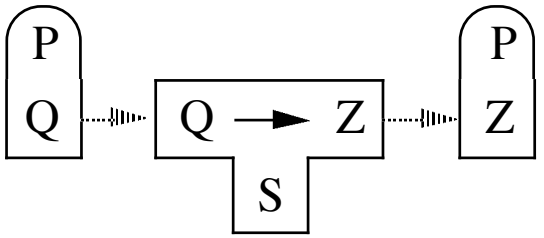
Themen

- Compiler
- Compiler-Generator
- Interpreter
- Compiler-Interpreter

Compiler

Ein Compiler ist ein Programm, das Programme einer Quellsprache Q in semantisch äquivalente Programme einer Zielsprache Z übersetzt





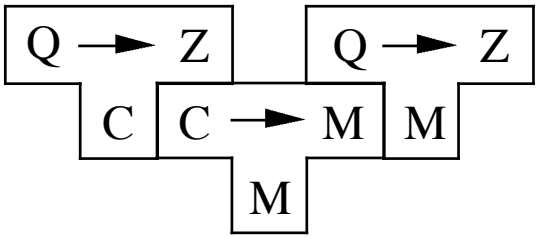
T–Diagramm

Das T–Diagramm enthält die Implementierungsprache S

Q, Z und S sind beliebige Programmiersprachen

Implementierungsprache

Ein Compiler für Q kann in einer höheren Sprache (z.B. C) implementiert werden, wenn es einen Compiler für diese Sprache gibt

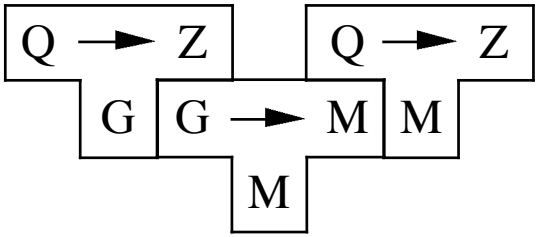


Varianten für Z

- **Maschinensprache M**
- **Beliebige andere Maschinensprache M' (Cross Compiler)**
- **Beliebige höhere Programmiersprache (Source-to-Source Compiler)**

Compiler-Generator

Ein Compiler-Generator erzeugt einen Compiler aus einer Spezifikation, formuliert in einer Compiler-Beschreibungsprache bzw. Generatorsprache G

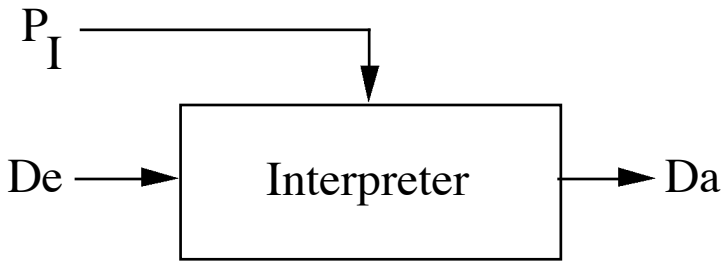


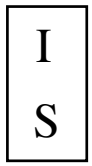
In der Praxis werden i.a. "syntaxgesteuerte" Compiler-Generatoren verwendet

Generatorsprache = Grammatik + Programmiersprache

Interpreter

Ein Interpreter (Abstrakte Maschine, Virtuelle Maschine) ist ein Programm, das Programme ausführen kann, die in der Interpretersprache I formuliert sind.





I–Diagramm

I und S sind beliebige Programmiersprachen

Compiler vs. Interpreter

Compiler analysiert und übersetzt das Programm einmal vor der Ausführung

Interpreter analysiert jede Anweisung jedesmal während der Ausführung
–> erhöhte Laufzeit bei Programmschleifen (ca. Faktor 10)

Vorteile des Compilers

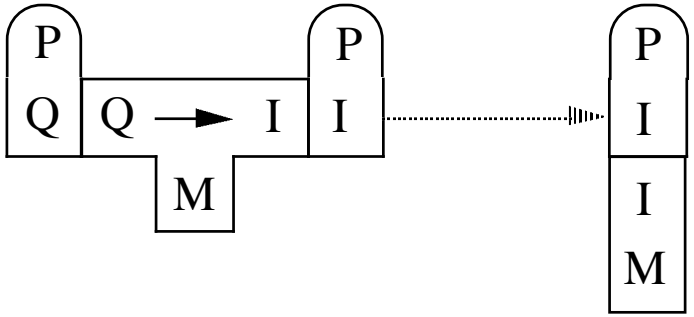
Statische Typüberprüfung, effiziente Zielprogramme

Vorteile des Interpreters

Programmänderungen zur Laufzeit, einfache Implementierung

Compiler–Interpreter System

Ein Compiler–Interpreter System übersetzt die Quellprogramme in eine Zwischensprache I, die interpretiert wird



Anforderungen an die Zwischensprache

- **Hardware–unabhängig**
- **einfache Erzeugung**
- **einfache Interpretation**

Beispiele: Pascal P–Code (1970), Java VM–Code (1995)

Postfix-Code

Infix-Ausdruck

```
3 * (5 + 7)
```

Postfix-Ausdruck

```
3 5 7 + *
```

Postfix-Code

```
push 3
push 5
push 7
add
mul
```

Stack (bei der Ausführung)

```
|-->
 3
 3   5
 3   5   7
 3  12
36
```

Postfix-Code als Zwischensprache für Compiler-Interpreter gut geeignet