

Compiler

Der Compiler ist in drei Phasen unterteilt. Der Scanner, welcher die Codesequenzen ihrer jeweiligen Eigenschaft zuweist, und der Parser, der mit den, vom Scanner erhaltenen, Informationen weiterverarbeitet, gehören zum Frontend des Compilers, während der Codegenerator zum Backend gehört.

Während der Scanner und der Parser für die jeweils zu übersetzende Sprache nur einmal implementiert werden müssen, existieren aufgrund der verschiedenen Eigenheiten der Betriebssysteme, n Codegeneratoren.

Scanner

Der Scanner erkennt in einem Quelltext die Terminalsymbole. Zu dieser Menge gehören Bezeichner, Zahlen und ein Vokabular. Dieses muss als erstes, noch bevor die erste Implementierung erfolgt festgelegt werden.

Erkannte Symbolwerte werden auf ganze Zahlen abgebildet, die durch die Konstantendefinition festgelegt werden.

Bei diesem Verfahren werden zwei Prozeduren aufgerufen. Die Prozedur *Mark* dient zum Erkennen von Fehlern. Wurde ein Fehler gefunden wird dessen Textposition im Quelltext markiert. Und die *Get*. Deren Aufgabe ist es, den Code in einzelne Elemente zu unterteilen und diese mit der jeweils erkannten Bedeutung zurück zu geben.

Parser

Anhand der Definition der Anfangs- und Folgesymbole, erkennt der Parser die vom Scanner erhaltenen Befehle. Die erkannte Syntax wird mittels rekursiven Abstiegs verarbeitet.

Des Weiteren muss der Parser syntaktische Fehler erkennen. Diese sind zum einen fehlerhafte- und zum anderen fehlende Symbole. Beide Fehler werden markiert. Der Parser darf bei erkannten Fehlern nicht gleich abbrechen, sondern sollte bei der nächsten Stelle, die er als korrekt erachtet, weiterarbeitet. Dadurch wird erreicht, dass in einem Durchlauf so viele Fehler wie möglich erkannt werden.

Codegenerator

Der Codegenerator besitzt zwei Arten der Codeerzeugung, die Direkte und die Verzögerte.

Das Problem bei der direkten Codeerzeugung ist, dass auf die einzelnen Register nicht mehr zugegriffen werden konnte. Diese einfache Codegenerierung erzeugt zwar korrekten, aber nicht optimalen Code. So müssen zum Beispiel Konstanten nicht in ein Register geladen werden, sondern können direkt zu Registerwerten addieren können. Dies wird mit der verzögerten Codegenerierung bewerkstelligt. Dessen Vorgehen besteht darin, die Information direkt dem Konstrukt hinzuzugeben und nicht zuerst als Code auszugeben. Die Codeausgabe ist nicht nur von den syntaktisch zu reduzierenden Symbolen abhängig, sondern auch von deren Attributen.