

Inhalt

1. Warum Verifizieren von Programmen?	1
1.1 Korrektheit von Programmen	2
1.2 Verifizieren versus Testen von Programmen	3
1.3 Programme gleichzeitig entwickeln und verifizieren	4
1.4 Für und wider Verifizieren	5
2. Einführende Beispiele	6
2.1 Mischen zweier sortierter Karteikartenstapel	6
2.2 Diskussion der Art der Präsentation des Verfahrens	9
2.3 Sortieren durch Mischen	12
2.4 Invarianten als Lösung von Rätselaufgaben	15
2.4.1 Weinpantschen	15
2.4.2 Hofstadters MU-Rätsel	16
3. Zusicherungen (Assertions)	19
3.1 Zusicherungen als Dokumentation von Programmen	19
3.2 Die Sprache der Zusicherungen	21
3.2.1 Zusicherungen als Boolesche Ausdrücke	22
3.2.2 Die Zusicherungen true und false	23
3.2.3 Zusicherungen mit Quantoren	24
3.2.4 Selbstdefinierte Prädikate	28
3.2.5 Selbstdefinierte arithmetische Funktionen	29
4. Programmzustände und Zustandsraum	31
4.1 Der Zusammenhang zwischen Zuständen und Zusicherungen	32
4.2 Programme als Abbildungen im Zustandsraum	34
5. Spezifizieren von Programmen	36
5.1 Spezifizieren mit Pre- und Postcondition	36
5.2 Beispiele für Spezifikationen	37
6. Verifikationsregeln (Verification rules)	49
6.1 Konsequenz-Regeln	50
6.2 Die Zuweisung	54
6.2.1 Die Zuweisung an eine einfache Variable	54
6.2.2 Die Mehrfachzuweisung	55
6.3 Die Sequenz	56
6.4 Die Alternative (if-Anweisung)	59
6.5 Die Iteration (Schleife)	61
6.5.1 Die while-Schleife	61
6.5.2 Die repeat-Schleife	67
6.6 Termination von Schleifen	72
6.7 Verifikation der while-Schleife	75

7. Entwickeln von Schleifen	77
7.1 Entwickeln einer Schleife aus einer gegebenen Invariante und Terminationsfunktion	77
7.2 Entwickeln von Invarianten aus gegebenen Spezifikationen	78
7.2.1 Die Invariante als Verallgemeinerung der Postcondition	78
7.2.2 Weglassen einer Bedingung	80
7.2.3 Konstante durch Variable ersetzen und Bereich der Variablen angeben	84
7.2.4 Kombinieren von Pre- und Postcondition	88
8. Die schwächste Precondition (weakest precondition)	90
8.1 Verifikation mit wp	93
8.2 Die wp der einzelnen Anweisungen	94
8.2.1 wp der Leeraanweisung	94
8.2.2 wp der Zuweisung	94
8.2.3 wp der Mehrfachzuweisung	95
8.2.4 wp der Sequenz	96
8.2.5 wp der Alternative	97
8.2.6 wp der Iteration	97
8.3 Die wp als Prädikatenstransformation	100
8.4 Definition der Semantik mit Hilfe der schwächsten Precondition	100
8.5 Eigenschaften der wp	101
8.6 Die schwächste Precondition für die Termination	103
9. Beispiele für Programmentwicklungen	106
9.1 Sortieren von Feldern	106
9.1.1 Sortieren durch direktes Einfügen	108
9.1.2 Sortieren durch Minimumsuche	109
9.2 Binäre Suche in einem Feld	110
9.3 Die Datenstruktur Heap (Halde)	112
9.4 Heapsort	119
9.5 Die M kleinsten Elemente von N Elementen	120
9.6 Maximaler Kursgewinn bei Wertpapieren	126
10. Unterprogramme (Prozeduren)	130
10.1 Die Prozedurdeklaration	130
10.2 Der Prozeduraufruf	132
10.3 Die schwächste Precondition des Prozeduraufrufs	133
10.4 Verifikation des Prozeduraufrufs	134
10.5 Spezifikation des Prozedurrumpfes mit externen Variablen	136
10.6 Verwendung von Variablen-Parametern	137
10.7 Eine verallgemeinerte Konsequenz-Regel	138

11. Invertieren von Programmen	142
11.1 Beispiele für inverse Anweisungen	143
11.2 Invertieren der zusammengesetzten Anweisung	144
11.3 Invertieren der alternativen Anweisung	144
11.4 Invertieren von Schleifen	145
11.5 Eine Analogie zum täglichen Leben	148
12. Parallele Programme	149
12.1 Zusammensetzen von parallelen Programmen	150
12.2 Beispiele für parallele Programme	151
12.2.1 Summe der Elemente eines Feldes	151
12.2.2 Skalarprodukt zweier Vektoren	156
12.2.3 Matrixmultiplikation	156

Anhang

A. Lösungen der Aufgaben	158
B. Syntax der Zusicherungen	172
C. Verifikationsregeln und wp	174
D. Aufwand von Verfahren	177

Literaturverzeichnis	179
Sachverzeichnis	181