

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	High-Level Programming Languages	1
1.2	Implementation of Programming Languages	2
1.2.1	Interpreters	2
1.2.2	Compilers	3
1.2.3	Real and Virtual Machines	4
1.2.4	Combined Compilation and Interpretation	4
1.3	General References	5
<b>2</b>	<b>Imperative Programming Languages</b>	<b>7</b>
2.1	Language Concepts and Their Compilation	7
2.2	The Architecture of the C-Machine	8
2.3	Simple Expressions and Assignments	9
2.4	Statements and Statement Sequences	15
2.5	Conditional and Iterative Statements	16
2.6	Memory Allocation for Variables of Basic Types	22
2.7	Memory Allocation for Arrays and Structures	23
2.8	Pointers and Dynamic Memory Allocation	27
2.9	Functions	32
2.9.1	Memory Organization of the C-Machine	35
2.9.2	Dealing with Local Variables	37
2.9.3	Function Call and Return	40
2.10	Translation of Programs	45
2.11	Exercises	48
2.12	List of CMA Registers	55
2.13	List of Code Functions of the CMA	55
2.14	List of CMA Instructions	55
2.15	References	55

<b>3</b>	<b>Functional Programming Languages</b>	57
3.1	Basic Concepts and Introductory Examples	57
3.2	A Simple Functional Programming Language	59
3.3	The Architecture of the MAMA	63
3.4	Translation of Simple Expressions	66
3.5	Access to Variables	68
3.6	<i>let</i> Expressions	73
3.7	Function Definitions	74
3.8	Function Application	76
3.9	Under- and Oversupply with Arguments	80
3.10	Recursive Variable Definitions	83
3.11	Closures and Their Evaluation	86
3.12	Optimization I: Global Variables	89
3.13	Optimization II: Closures	90
3.14	Translating Program Expressions	91
3.15	Structured Data	92
3.15.1	Tuples	92
3.15.2	Lists	93
3.15.3	Closures for Tuples and Lists	96
3.16	Optimization III: Last Calls	97
3.17	Exercises	99
3.18	List of MAMA Registers	103
3.19	List of Code Functions of the MAMA	103
3.20	List of MAMA Instructions	104
3.21	References	104
<b>4</b>	<b>Logic Programming Languages</b>	105
4.1	The Language PROL	105
4.2	The Architecture of the WIM	108
4.3	Allocation of Terms in the Heap	109
4.4	The Translation of Literals	114
4.5	Unification	115
4.6	Clauses	126
4.7	The Translation of Predicates	127
4.7.1	Backtracking	127
4.7.2	Putting It All Together	130
4.8	The Finalization of Clauses	131
4.9	Queries and Programs	134
4.10	Optimization I: Last Goals	135
4.11	Optimization II: Trimming of Stack Frames	138
4.12	Optimization III: Clause Indexing	140
4.13	Extension: The Cut Operator	142
4.14	Digression: Garbage Collection	144
4.15	Exercises	149
4.16	List of WIM Registers	152

4.17	List of Code Functions of the WiM .....	152
4.18	List of WiM Instructions .....	152
4.19	References .....	153
<b>5</b>	<b>Object-Oriented Programming Languages .....</b>	<b>155</b>
5.1	Concepts of Object-Oriented Languages .....	155
5.1.1	Objects .....	155
5.1.2	Object Classes .....	156
5.1.3	Inheritance .....	157
5.1.4	Genericity .....	158
5.1.5	Information Encapsulation .....	158
5.1.6	Summary .....	159
5.2	An Object-Oriented Extension of C .....	159
5.3	The Memory Organization for Objects .....	160
5.4	Method Calls .....	163
5.5	The Definition of Methods .....	165
5.6	The Use of Constructors .....	166
5.7	The Definition of Constructors .....	168
5.8	Perspective: Multiple Inheritance .....	169
5.9	Exercises .....	171
5.10	List of Additional Registers .....	177
5.11	CMa Instructions for Objects .....	177
5.12	References .....	177
	<b>References .....</b>	<b>179</b>
	<b>Index .....</b>	<b>183</b>



<http://www.springer.com/978-3-642-14908-5>

Compiler Design

Virtual Machines

Wilhelm, R.; Seidl, H.

2010, XIII, 187 p., Hardcover

ISBN: 978-3-642-14908-5