

# Contents

<b>1</b>	<b>Processes</b>	<b>2</b>
1.1	Definitions and notation . . . . .	2
1.2	The value-passing calculus . . . . .	7
1.3	Renaming . . . . .	10
<b>2</b>	<b>Equivalence of processes</b>	<b>12</b>
2.1	Strong bisimulation and strong equivalence . . . . .	12
2.2	Observable bisimulation . . . . .	18
2.3	Equivalence checking, solving equations . . . . .	21
<b>3</b>	<b>Hennessy–Milner logic</b>	<b>27</b>
3.1	Basic notions and definitions . . . . .	27
3.2	Connecting the structure of actions with modal properties . . . . .	30
3.3	Observable modal logic . . . . .	32
3.4	Necessity and divergence . . . . .	34
<b>4</b>	<b>Alternative characterizations of process equivalences</b>	<b>35</b>
4.1	Game semantics . . . . .	35
4.2	Weak bisimulation properties . . . . .	38
4.3	Modal properties and equivalences . . . . .	38
<b>5</b>	<b>Temporal logics for processes</b>	<b>41</b>
5.1	Temporal behaviour of processes . . . . .	41
5.2	Linear time logic . . . . .	43
5.3	Model checking in linear time logic . . . . .	45
5.3.1	Linear time properties . . . . .	45
5.3.2	Towards LTL model checking . . . . .	49
5.3.3	Büchi automata and model checking for LTL formulas . . . . .	63
5.3.4	Complexity of the LTL model checking process . . . . .	67
5.3.5	Fairness assumptions in LTL . . . . .	67
<b>6</b>	<b>Computation Tree Logic</b>	<b>69</b>
6.0.6	Syntax of CTL . . . . .	69
6.0.7	Semantics of CTL . . . . .	70
6.0.8	Normal forms and expressiveness . . . . .	73
6.0.9	Relating CTL with LTL . . . . .	75
6.1	Model checking in CTL . . . . .	76

6.1.1	The crucial idea . . . . .	76
6.1.2	Searches based on the expansion laws . . . . .	78
6.1.3	Symbolic model checking . . . . .	80
6.1.4	Binary decision diagrams . . . . .	83
6.2	CTL* . . . . .	90
6.2.1	Basic notions and definitions . . . . .	90
6.2.2	Model checking in CTL* . . . . .	90
<b>7</b>	<b>Hennessey–Milner logic with recursive equations</b>	<b>92</b>
7.1	Modal formulas with variables . . . . .	92
7.2	Syntax and semantics of Hennessey–Milner logic with recursion . . . . .	94
7.3	The correctness of CTL-satisfiability algorithms . . . . .	97
7.4	Equational systems with several recursive variables . . . . .	99
7.5	Largest and least fixpoints mixed . . . . .	101
<b>8</b>	<b>The modal <math>\mu</math>-calculus</b>	<b>104</b>
8.1	Logic and fixpoints . . . . .	104
8.2	Playing with modal $\mu$ -formulas . . . . .	106
8.3	Game characterization of satisfiability of modal $\mu$ -formulas . . . . .	107
<b>9</b>	<b>Conclusions</b>	<b>112</b>
9.1	General overview of the approaches discussed . . . . .	112
9.2	Model checking in practice . . . . .	113
<b>10</b>	<b>Appendix: The programming language Erlang</b>	<b>114</b>
10.1	Introduction . . . . .	114
10.2	Datatypes in Erlang . . . . .	114
10.2.1	Pattern matching . . . . .	116
10.2.2	Defining functions . . . . .	117
10.2.3	Conditional expressions . . . . .	118
10.2.4	Scope of variables . . . . .	120
10.2.5	The module system . . . . .	121
10.3	Programming with recursion . . . . .	121
10.3.1	Programming with lists . . . . .	121
10.3.2	Programming with tuples . . . . .	129
10.3.3	Records . . . . .	130
10.4	Concurrent processes . . . . .	133
10.4.1	Creating processes . . . . .	133
10.4.2	Simple examples . . . . .	134
10.4.3	Handling timeouts . . . . .	139
10.4.4	Registered processes . . . . .	141
10.5	Distributed programming . . . . .	142
	<b>Bibliography</b>	<b>146</b>