

Contents

1	Introduction	1
2	Preliminaries	5
2.1	Machine learning	5
2.1.1	What does 'learning' mean?	8
2.1.2	The learning framework	10
2.1.3	Learning as heuristic search	12
2.1.4	Dimensions of learning	16
2.1.5	Evolutionary algorithms	19
2.1.6	The real-world learning scenario	26
2.2	Formal language theory	30
2.2.1	Context-free grammars	31
2.2.2	Derivation trees	33
2.2.3	Attribute grammars	35
2.2.4	Decorated derivation trees	39
3	Attribute Grammar Based Genetic Programming	47
3.1	Biasing genetic programming	48
3.1.1	Dealings with erroneous individuals	50
3.1.2	Syntactical constraints as language bias	53
3.2	Using attribute grammars to describe validation constraints	57
3.2.1	Augmenting DTGP	58
3.2.2	Usage of attributes	59
3.3	Random generation of decorated derivation trees	60
3.3.1	How to avoid the bloating of derivation trees	61
3.3.2	Minimal derivation trees	63
3.3.3	Enhanced random tree generation	65
3.3.4	Calculating minimal derivation trees	66
3.4	Issues concerning appropriate randomization	72
3.4.1	Random rule selection	72
3.4.2	Random boundary on tree growth	73
3.5	Random node selection	76
3.6	Evolutionary operators	79

3.6.1	Mutation	80
3.6.2	Crossover	81
3.6.3	Global shuffle-crossover	82
3.6.4	Biasing the creation and the selection of pools	83
3.7	Effectively evolvable attribute grammars	86
3.7.1	Invalid decorated derivation trees	89
3.7.2	Partially valid derivation trees	93
4	Evolving redundancy-free FIR structures	97
4.1	The design of digital filters	98
4.1.1	Overview of the chip design flow	99
4.1.2	Description of digital filters	100
4.1.3	Polynomial FIR structures	103
4.1.4	Graphical representation of FIR structures	104
4.2	The <i>evolFIR</i> system	106
4.3	Representing FIR structures by derivation trees	108
4.3.1	Syntactical description of function blocks	108
4.3.2	Attribution of derivation trees	109
4.4	Generation of valid derivation trees	111
4.4.1	General constraints on the decomposition of the transfer function	111
4.4.2	Decomposition via addition	114
4.4.3	Decomposition via delay	116
4.4.4	Decomposition via shift	117
4.4.5	Decomposition via multiplication	117
4.5	Eliminating redundancy	120
4.5.1	Abstract derivation tree	120
4.5.2	Abstract-linked derivation tree	120
4.6	Evolutionary operators for abstract-linked trees	124
4.6.1	Mutation of abstract-linked subtrees	125
4.6.2	Crossing of abstract-linked subtrees	129
4.6.3	Fitness function	133
4.7	Evolutionary engine	133
4.8	Generating the VHDL descriptions	134
4.8.1	Pipelining and retiming	136
4.9	Recently used techniques in circuit design	138
5	Experimental results	141
5.1	Evolutionary parameter-group	142
5.1.1	Comma-strategy model of evolutionary engine	144
5.1.2	Testing evolutionary operators individually	145
5.1.3	Combining crossover and mutation	150
5.1.4	Plus-strategy model of evolutionary engine	152

5.1.5	The impact of the selection methods	153
5.1.6	The impact of quantitative alterations of the settings . . .	157
5.1.7	Best setting for the evolutionary parameters	160
5.2	Hardware-specific parameter-group	160
5.2.1	Hardware constraints	160
5.2.2	Hardware optimization parameters	162
5.2.3	Biasing the application of grammar rules	163
5.2.4	Altering the application of grammar rules	166
5.2.5	Best setting for the hardware optimization parameters . .	170
5.3	Summary	170
6	Conclusion and Future Work	173
6.1	Conclusion	173
6.2	Future Work	176
A	The transfer function used to test the <i>evolFIR</i> system	179
B	Command-line parameters of the <i>evolFIR</i> system	181
C	Example XML output of <i>evolFIR</i>	187
D	Example VHDL description generated from the XML output of <i>evolFIR</i>	191
	Bibliography	193
	Nomenclature	201
	Index	205
	List of Figures	209
	List of Tables	213