

Contents

1	Proof Support for General Type Classes	1
	<i>Ron van Kesteren, Marko van Eekelen, Maarten de Mol</i>	
1.1	Introduction	4
1.2	Sparkle	5
1.3	Preliminaries	7
1.4	Structural induction	8
1.5	Induction on instances	10
1.6	Multiple class constraints	14
1.7	Implementation	14
1.8	Related and future work	15
1.9	Conclusion	15
	References	15
2	Generic Proofs for Combinator-based Generic Programs	17
	<i>Fermín Reig</i>	
2.1	Introduction	19
2.2	‘Boilerplate’ combinators as generic functions	20
2.3	Generic proofs for ‘boilerplate’ combinators	21
2.4	A proof of the fusion law for <i>gmapT</i>	22
2.5	A theorem about <i>occurs</i>	25
2.6	A fusion law for <i>everywhere</i>	28
2.7	Conclusions and further work	29
2.8	Proof of lemma 2.1	29
2.9	Proof of lemma 2.2	30
2.10	Proof of lemma 2.3	30
2.11	Proof of theorem 2.5 (continued)	31
	References	31
3	Building certified components within FOCAL	33
	<i>Catherine Dubois, Thérèse Hardin, Véronique Viguié Donzeau Gouge</i>	
3.1	Introduction	34
3.2	FOCAL requirements and design principles	35
3.3	Overview of FOCAL	35

3.4	A complete example	38
3.5	Compiling	43
3.6	Applications	45
3.7	Related work	46
3.8	Conclusion and future work	46
	References	47
4	Calculating an Exceptional Machine	49
	<i>Graham Hutton, Joel Wright</i>	
4.1	Introduction	49
4.2	Abstract machines	50
4.3	Arithmetic expressions	51
4.4	Adding exceptions	58
4.5	Further work	63
	References	64
5	Generalizing the AUGMENT Combinator	65
	<i>Neil Ghani, Tarmo Uustalu, Varmo Vene</i>	
5.1	Introduction	65
5.2	Semantics of FOLD/BUILD	66
5.3	The AUGMENT of free monads	69
5.4	A generalized AUGMENT combinator	72
5.5	Conclusion and future work	77
	References	77
6	Alice Through the Looking Glass	79
	<i>Andreas Rossberg, Didier Le Botlan, Guido Tack, Th. Brunklaus, Gert Smolka</i>	
6.1	Introduction	79
6.2	Futures	80
6.3	Higher-order modules	83
6.4	Packages	84
6.5	Components	86
6.6	Decomposing components	88
6.7	Distribution	89
6.8	Implementation	92
6.9	Related work	92
6.10	Outlook	93
	References	94
7	Experiments with GHC's Optimiser	97
	<i>László Németh</i>	
7.1	Introduction	97
7.2	Contributions of the paper	98
7.3	The setup	98
7.4	The method	99

7.5	Evaluation	102
7.6	Related work	106
7.7	Conclusions	107
	References	108
8	Disjoint Forms in Graphical User Interfaces	113
	<i>Sander Evers, Peter Achten, Rinus Plasmeijer</i>	
8.1	Introduction	113
8.2	FunctionalForms summary	114
8.3	Combinators for disjoint forms	116
8.4	Implementation	119
8.5	Safety	125
8.6	Related work	126
8.7	Conclusions and future work	126
	References	127
9	A Graphic Functional-Dataflow Language	129
	<i>Silvia Clerici, Cristina Zoltan</i>	
9.1	Introduction	129
9.2	Outline of the NiMo language	131
9.3	The NiMo environment	138
9.4	Current state and future work	142
9.5	Concluding remarks	143
	References	144