

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>11</b>
2.1	Gradual Typing . . . . .	11
2.2	Abstracting Gradual Typing . . . . .	15
2.3	Complex Type Disciplines and Language Constructs of Interest . . . . .	16
2.3.1	References . . . . .	17
2.3.2	Information-Flow Security Typing . . . . .	19
2.3.3	Parametric Polymorphism . . . . .	22
<b>3</b>	<b>First step: Gradualizing References</b>	<b>25</b>
3.1	Gradual Typing with References . . . . .	25
3.2	Preliminary: The Static Language $\lambda^{\text{REF}}$ . . . . .	27
3.3	Gradualizing $\lambda^{\text{REF}}$ . . . . .	29
3.3.1	Syntax and Meaning of Gradual Types . . . . .	29
3.3.2	Lifting the Type System . . . . .	31
3.3.3	Static Semantics . . . . .	32
3.3.4	Dynamic Semantics . . . . .	33
3.3.5	Elaboration of $\widetilde{\lambda}_{\varepsilon}^{\text{REF}}$ terms . . . . .	38
3.3.6	Properties . . . . .	40
3.3.7	$\widetilde{\lambda}^{\text{REF}}$ in Action . . . . .	41
3.4	Comparing $\widetilde{\lambda}^{\text{REF}}$ and HCC . . . . .	42
3.4.1	The Coercion Calculus . . . . .	42
3.4.2	Relating $\widetilde{\lambda}_{\varepsilon}^{\text{REF}}$ and HCC <sup>+</sup> . . . . .	47
3.4.3	Recovering Space Efficiency in $\widetilde{\lambda}^{\text{REF}}$ . . . . .	50
3.5	Encoding Permissive and Monotonic References in $\widetilde{\lambda}^{\text{REF}}$ . . . . .	52
3.5.1	Static Semantics . . . . .	53
3.5.2	Dynamic Semantics . . . . .	53
3.5.3	Properties . . . . .	54
3.6	Related work . . . . .	55
3.7	Conclusion . . . . .	56
<b>4</b>	<b>Type-driven Gradual Security Typing</b>	<b>58</b>
4.1	Introduction . . . . .	59
4.2	Type-Driven Gradual Security Typing in Action . . . . .	61
4.3	Static Security Typing with References . . . . .	65

4.4	GSL <sub>Ref</sub> : Type-Driven Gradual Security Typing . . . . .	69
4.4.1	Static Semantics . . . . .	72
4.4.2	Dynamic Semantics . . . . .	73
4.4.3	Examples of Reduction . . . . .	83
4.4.4	GSL <sub>Ref</sub> : Safety and Graduality . . . . .	84
4.4.5	Prototype Implementation . . . . .	86
4.5	GSL <sub>Ref</sub> : Noninterference . . . . .	86
4.6	Deriving GSL <sub>Ref</sub> with AGT (almost) . . . . .	91
4.6.1	Deriving the Statics . . . . .	92
4.6.2	Deriving the Dynamics . . . . .	93
4.6.3	Policing Dynamic Heap Updates . . . . .	97
4.7	Related Work . . . . .	99
4.8	Conclusion . . . . .	105
<b>5</b>	<b>A Gradual Interpretation of Union Types</b>	<b>106</b>
5.1	Introduction . . . . .	106
5.2	Background and Motivation . . . . .	108
5.2.1	Tagged Unions . . . . .	108
5.2.2	Untagged Unions . . . . .	109
5.2.3	Gradual Unions . . . . .	110
5.2.4	Comparing Unions . . . . .	111
5.2.5	Gradual Unions vs. Standard Gradual Types . . . . .	113
5.3	GTFL <sup>⊕</sup> : Static Semantics . . . . .	114
5.3.1	The Static Language: STFL . . . . .	114
5.3.2	Defining Gradual Types Separately . . . . .	114
5.3.3	Combining Gradual Types: Take 1 . . . . .	116
5.3.4	Combining Gradual Types: Take 2 . . . . .	117
5.3.5	Static Semantics of GTFL <sup>⊕</sup> . . . . .	120
5.4	GTFL <sup>⊕</sup> : Dynamic Semantics and Properties . . . . .	122
5.5	Correctness of the Translational Semantics . . . . .	125
5.6	Related Work . . . . .	129
5.7	Conclusion . . . . .	130
<b>6</b>	<b>Gradual Parametricity, Revisited</b>	<b>132</b>
6.1	Introduction . . . . .	132
6.2	The Need to Revisit Gradual Parametricity . . . . .	133
6.2.1	Static Semantics Issues . . . . .	134
6.2.2	Dynamic Semantics Issues . . . . .	135
6.3	GSF, Informally . . . . .	137
6.3.1	Design Principles, Goals and Non-Goals . . . . .	137
6.3.2	GSF in Action . . . . .	138
6.4	Preliminary: The Static Language SF . . . . .	139
6.5	GSF: Statics . . . . .	141
6.5.1	Syntax and Syntactic Meaning of Gradual Types . . . . .	141
6.5.2	Lifting the Static Semantics . . . . .	143
6.5.3	Static Properties of GSF . . . . .	145
6.6	GSF: Evidence-Based Dynamics . . . . .	145

6.7	Evidence for Gradual Parametricity . . . . .	148
6.7.1	Simple Evidence, and Why It Fails . . . . .	148
6.7.2	Refining Evidence . . . . .	149
6.7.3	Basic Properties of GSF Evaluation . . . . .	152
6.8	GSF: Parametricity . . . . .	153
6.9	Parametricity vs. Dynamic Gradual Guarantee . . . . .	155
6.10	Gradual Free Theorems in GSF . . . . .	157
6.11	Related Work . . . . .	159
6.12	Conclusion . . . . .	160
<b>7</b>	<b>Conclusions and Future Work</b>	<b>162</b>
	<b>Bibliography</b>	<b>165</b>
	<b>Appendices</b>	<b>176</b>
<b>Appendix A</b>	<b>First step: Gradualizing References</b>	<b>177</b>
A.1	Gradualizing $\lambda^{\text{REF}}$ , Elaborating $\widetilde{\lambda}_\varepsilon^{\text{REF}}$ . . . . .	177
A.2	Type Safety . . . . .	180
A.3	Gradual Guarantee . . . . .	184
A.3.1	Conservative Extensions of the Static Discipline . . . . .	184
A.3.2	Static Gradual Guarantee . . . . .	193
A.3.3	Dynamic Gradual Guarantee . . . . .	196
A.4	Relation to the coercion calculus . . . . .	201
A.5	Encoding Permissive and Monotonic References in $\widetilde{\lambda}^{\text{REF}}$ . . . . .	215
<b>Appendix B</b>	<b>Type-driven Gradual Security Typing</b>	<b>219</b>
B.1	Additional Definitions . . . . .	219
B.1.1	$\text{SSL}_{\text{Ref}}$ : Static Semantics . . . . .	219
B.1.2	$\text{SSL}_{\text{Ref}}$ : Noninterference Definitions . . . . .	219
B.1.3	$\text{GSL}_{\text{Ref}}$ : Static Semantics . . . . .	222
B.1.4	$\text{GSL}_{\text{Ref}}^\varepsilon$ : Static Semantics . . . . .	223
B.1.5	$\text{GSL}_{\text{Ref}}^\varepsilon$ : Dynamic Semantics . . . . .	224
B.1.6	$\text{GSL}_{\text{Ref}}$ : Translation to $\text{GSL}_{\text{Ref}}^\varepsilon$ . . . . .	227
B.2	Static Security Typing with References . . . . .	229
B.2.1	$\text{SSL}_{\text{Ref}}$ : Static type safety . . . . .	229
B.2.2	$\text{SSL}_{\text{Ref}}$ : Noninterference . . . . .	240
B.3	Gradualizing the Static Semantics . . . . .	251
B.3.1	From Gradual Labels to Gradual Types . . . . .	251
B.3.2	Static Criteria for Gradual Typing . . . . .	253
B.4	Gradualizing the Dynamic Semantics . . . . .	259
B.4.1	Precise Evidence for Consistent Security Judgments . . . . .	259
B.4.2	Initial evidence . . . . .	261
B.4.3	Evolving evidence: Consistent Transitivity . . . . .	262
B.4.4	Algorithmic definitions . . . . .	263
B.4.5	Proofs . . . . .	266
B.5	$\text{GSL}_{\text{Ref}}^\varepsilon$ : Dynamic properties . . . . .	271

B.5.1	Intrinsic Terms: Static Semantics . . . . .	271
B.5.2	Intrinsic Terms: Dynamic Semantics . . . . .	274
B.5.3	Relating Intrinsic and Evidence-augmented Terms . . . . .	275
B.5.4	Type Safety . . . . .	279
B.5.5	Dynamic Gradual Guarantee . . . . .	289
B.5.6	Noninterference . . . . .	296
<b>Appendix C</b>	<b>A Gradual interpretation of Union Types</b>	<b>320</b>
C.1	Gradual Unions . . . . .	320
C.1.1	The Static Language: STFL . . . . .	320
C.1.2	Syntax and Meaning of Gradual Unions . . . . .	325
C.1.3	Static Semantics of $\text{GTFL}^\oplus$ . . . . .	327
C.1.4	Dynamic Semantics of $\text{GTFL}^\oplus$ . . . . .	334
C.1.5	Properties of the Gradual Security Language . . . . .	335
C.2	Compiling $\text{GTFL}^\oplus$ to Threesomes . . . . .	344
C.2.1	Intermediate language: $\text{GTFL}^\oplus_{\rightarrow}$ . . . . .	344
C.2.2	Cast Insertion . . . . .	345
C.2.3	Correctness of the Translational Semantics . . . . .	348
<b>Appendix D</b>	<b>Gradual Parametricity, Revisited</b>	<b>357</b>
D.1	SF: Well-formedness . . . . .	357
D.2	GSF: Statics . . . . .	358
D.2.1	Syntax and Syntactic Meaning of Gradual Types . . . . .	358
D.2.2	Lifting the Static Semantics . . . . .	359
D.2.3	Well-formedness . . . . .	363
D.2.4	Static Properties . . . . .	363
D.3	GSF: Dynamics . . . . .	369
D.3.1	Evidence Type Precision . . . . .	369
D.3.2	Initial Evidence . . . . .	369
D.3.3	Consistent Transitivity . . . . .	369
D.3.4	$\text{GSF}_\varepsilon$ : Dynamic Semantics . . . . .	369
D.3.5	Translation from GSF to $\text{GSF}_\varepsilon$ . . . . .	371
D.4	GSF: Properties . . . . .	374
D.4.1	Type Safety . . . . .	374
D.4.2	Static Terms Do Not Fail . . . . .	378
D.5	GSF: Parametricity . . . . .	382
D.5.1	Auxiliary Definitions . . . . .	382
D.5.2	Fundamental Property . . . . .	382
D.5.3	Contextual Equivalence . . . . .	417
D.6	GSF: Imprecise Termination . . . . .	419
D.7	A Cheap Theorem in GSF . . . . .	427