

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Overview of Gradual Refinement Types . . . . .	3
1.2	Deriving Gradual Refinement Types . . . . .	4
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Language Semantics and Type Systems . . . . .	5
2.1.1	The Lambda Calculus . . . . .	5
2.1.2	Operational Semantics . . . . .	6
2.1.3	Type Systems . . . . .	7
2.2	Refinement Types . . . . .	10
2.2.1	Dependent Function Types . . . . .	11
2.2.2	Flow Sensitivity . . . . .	11
2.2.3	Subtyping and Verification Conditions . . . . .	12
2.3	Satisfiability Modulo Theory . . . . .	12
2.3.1	Formal Definition and Conventions . . . . .	14
2.3.2	Background Theories . . . . .	15
2.4	Gradual Typing . . . . .	16
2.4.1	Gradual Types . . . . .	17
2.4.2	Type Consistency . . . . .	17
2.4.3	Casts as Runtime Checks . . . . .	18
2.4.4	Type Precision . . . . .	19
2.5	Abstracting Gradual Typing . . . . .	19
2.5.1	Lifting Predicates to Gradual Types . . . . .	20
2.5.2	Lifting Functions on Gradual Types . . . . .	21
2.5.3	Deriving the Runtime Semantics . . . . .	22
<b>3</b>	<b>A Static Refinement Type System</b>	<b>24</b>
3.1	Syntax and Operational Semantics . . . . .	24
3.2	Type System . . . . .	26
3.2.1	Well-formedness . . . . .	27
3.2.2	Typing . . . . .	27
3.2.3	Subtyping . . . . .	30
3.3	Soundness . . . . .	30
<b>4</b>	<b>A Gradual Refinement Types System</b>	<b>32</b>
4.1	Gradual Formulas, Types and Environments . . . . .	32

4.1.1	Gradual Formulas . . . . .	32
4.1.2	Gradual Types . . . . .	33
4.1.3	Gradual Logical Environments . . . . .	33
4.2	Consistent Relations and Functions . . . . .	34
4.3	The System and its Properties . . . . .	35
<b>5</b>	<b>Defining the Logical Imprecision</b>	<b>37</b>
5.1	Naive Interpretation . . . . .	37
5.2	Non-Contradicting Interpretation . . . . .	38
5.3	Semantic Interpretation . . . . .	39
5.4	Local Interpretation . . . . .	40
5.5	Abstracting Formulas . . . . .	43
<b>6</b>	<b>Abstracting the Dynamic Semantics</b>	<b>46</b>
6.1	Evidence for Consistent Subtyping . . . . .	47
6.2	Consistent Subtyping Transitivity . . . . .	48
6.3	Consistent Subtyping Substitution . . . . .	49
6.4	Consistent Term Substitution . . . . .	51
6.5	Dynamic Semantics and Properties . . . . .	52
6.5.1	Intrinsic Terms . . . . .	52
6.5.2	Reduction . . . . .	52
6.5.3	Consistent Term Substitution . . . . .	56
6.5.4	Properties of the Gradual Refinement Types Language . . . . .	56
<b>7</b>	<b>Towards a Practical Implementation</b>	<b>58</b>
7.1	Algorithmic Consistent Type Substitution . . . . .	58
7.2	Algorithmic Consistent Subtyping . . . . .	59
7.3	A Boost in Expressiveness: Measures . . . . .	61
7.4	Revisiting the Dynamic Semantics . . . . .	64
<b>8</b>	<b>Conclusions</b>	<b>65</b>
	<b>Bibliography</b>	<b>67</b>