

A Language with Class

**The Theory of Classification Exemplified in
an Object-Oriented Language**

Anthony James Howard Simons

This thesis is submitted for the degree of
Doctor of Philosophy,

in the Department of Computer Science,
University of Sheffield,

December 1995.

A Language with Class

The Theory of Classification Exemplified in an Object-Oriented Programming Language

Anthony James Howard Simons

Abstract

Object-oriented programming languages have developed in advance of a formal theory of classification and, as a result, their treatment of *class* is muddled and ill-founded. The operational behaviour of such languages is often described incorrectly using a terminology of types which strictly does not apply. The type systems of languages affected by such misunderstandings are incorrect; furthermore, they may contain redundant mechanisms for handling type polymorphism as a result of the central ambiguity surrounding the notion of *class*. To counter this confusion, a mathematical theory of classification is developed, which encompasses other approaches to type abstraction, such as "type constructors", "generic parameters", "classes", "inheritance" and "polymorphism". The theory extends earlier work on F-bounded polymorphism, by providing second- and higher-order typed record combination for objects modelled as records of functions. The theory captures notions such as multiple and mixin inheritance and correctly classifies types with internal polymorphism. Recursive structures having both homogenous and heterogenous internal polymorphism are supported. To demonstrate that theoretical purity does not mitigate against practical concerns, the theory is exemplified in a programming language called *Brunel*. An innovation in *Brunel* is the treatment of all forms of polymorphism using a single parametric mechanism; polymorphism is resolved either statically at compile-time or dynamically at run-time, by an optimising compiler. F-bounded types, type constructors and simple types are all handled within the same framework, demonstrating that the mathematical notion of *class* is fundamental to all forms of type abstraction.

Acknowledgements

I am deeply indebted to the following people:

Phil Green, who changed me from a linguist into a computer scientist and gave me my first taste of object-orientation; Tony Cowling, Matt Fairtlough and Mike Holcombe, for fascinating conversations over coffee and standing in doorways about abstract type algebras and λ -calculus; Colin Smythe, for constant encouragement and supervisory guidance in bringing this project to completion;

William Cook, Peter Canning, Walt Hill, Walter Olthoff and John Mitchell, for original inspiration and insightful conversations at OOPSLA; Warren Harris, Kim Bruce, Luca Cardelli and Benjamin Pierce, for access to their work and many discussions via email or in person to clarify technical points;

Low Eng-Kwang, Tse Hau-Pui, Ong Pang-Siong, Ng Yee-Mei, David Black, Rex Tam and Mark Williams, for working under my direction to develop parts of the *Brunel* compiler and programming environment;

Penny Simons for the kind loan of a PC and many evenings and weekends; Charlotte Simons for being an amazingly patient daughter; and finally John and Elinor Simons, without whom none of this could have been achieved.

Contents

Title Page	i
Abstract.....	ii
Acknowledgements.....	iii
Contents.....	iv
1 Introduction.....	1
1.1 In Search of Class	1
1.2 A Guide to the Thesis.....	3
2 Views of Classification.....	6
2.1 The Object-Oriented Paradigm.....	6
2.2 An Object-Oriented Perspective	11
2.3 Defining 'Object-Oriented'	14
2.4 The Challenge of Classification.....	23
3 Object Types and Subtypes	30
3.1 Abstract Types	30
3.2 Object Types	34
3.3 A Calculus of Subtyping	39

3.4 Object Subtyping	47
4 Polymorphic Inheritance	52
4.1 Covariance and Contravariance	52
4.2 Inheritance as Subtyping	56
4.3 Bounded Universal Quantification	61
4.4 Function-Bounded Quantification	67
5 A Theory of Classification	76
5.1 Classification and Hierarchy	76
5.2 Classification and Derivation	85
5.3 Classification and Components	92
6 Multidimensional Classification	99
6.1 The Dimensions of Classification	99
6.2 Multiple Classification	105
6.3 Higher Classification	112
7 A Language with Objects	122
7.1 Object State	122
7.2 Object Creation	125
7.3 Object Identity	133
8 A Language with Class	143
8.1 Class Designs	143
8.2 Inheritance Designs	149
8.4 Polymorphic Designs	154
9 A Language with Flow	165
9.1 Selection and Iteration	165
9.2 Dynamic Dispatch and Mapping	171
9.3 Scope and Type Nondeterminism	180

10 Compiling a Language with Class	186
10.1 Introducing the Brunel Project	186
10.2 Compiling Object-Oriented Languages	188
10.3 Global Optimisation of Inheritance	193
10.4 Global Optimisation of Bindings	197
10.5 Code Generation Strategies	202
10.6 Summary of Optimisation Techniques.....	205
11 Conclusion	208
11.1 Summary of Findings	208
11.2 In Support of the Thesis	213
11.3 Further Work	217
Appendix 1	220
A.1 λ -Calculus	220
A.2 Models for Mathematics	223
A.3 Models for Data Structures	228
A.4 Extending λ -Calculus	235
Bibliographic References	239