The Malevolent Ingenuity of Preprocessor Metaprogramming in C

Temirkhan Myrzamadi

September 18, 2020

ii

Contents

0.1   Foreword

The macro system is probably the most contradictory feature of C: it provides a way to extend the host programming language with new syntactical constructs, thus forcing some ill-formed programs fail to compile, through is an increasingly treacherous technology, being able to potentially cause a numerous amount of bugs out of nothing.

This book was derived by my continuous experiments with the preprocessor: I have implemented some of the concepts of functional programming, including algebraic data types (ADTs) with exhaustive pattern matching, parametric polymorphism, higher-kinded types, generalised ADTs, as well as from the world of object orientation: classes, dynamic dispatch, and all the coherent stuff: type introspection, type-safe error handling facilities and more.

So what I want to say: writing macros is an art. Macros, overall, can be considered as computer languages on their own, with custom syntax, given by their signatures, and semantics, defined by their bodies. I believe that this book will be useful for everyone programming in C, directing a reader to much cleaner macro interfaces and their implementations.

0.2 Preface